



Fast Models

Version 11.27

Reference Guide

Non-Confidential

Copyright © 2017–2024 Arm Limited (or its affiliates). All rights reserved.

Issue 00

100964_1127_00_en



Fast Models Reference Guide

This document is Non-Confidential.

Copyright © 2017–2024 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_1127_00_en) was issued on 2024-09-16. There might be a later issue at <http://developer.arm.com/documentation/100964>

The product version is 11.27.

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.

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

1. About the models.....	22
1.1 Model capabilities.....	22
1.2 Running Arm Software Test Libraries (STL) on Fast Models.....	22
1.3 Quality level definitions.....	23
1.4 Fast Models accuracy.....	23
1.4.1 Timing accuracy of Fast Models.....	24
1.4.2 Bus traffic in Fast Models.....	24
1.4.3 Instruction prefetch in Fast Models.....	26
1.4.4 Out-of-order execution and write-buffers in Fast Models.....	26
1.4.5 Caches in Fast Models.....	27
1.4.6 Global exclusive monitor in Fast Models.....	28
1.5 Processor implementation.....	28
1.5.1 Mapping PMU events to MTI trace sources.....	28
1.5.2 Caches in PV models.....	30
1.5.3 GICv3 in PV models.....	32
1.5.4 GICv4 in PV models.....	32
1.5.5 CP14 Debug coprocessor.....	33
1.5.6 TLBs in PV models.....	33
1.5.7 Memory access in PV models.....	33
1.5.8 Timing in PV models.....	34
1.5.9 CADI interactions with processor behavior.....	35
1.6 Fast Models CADI implementation.....	36
1.7 CADI sync watchpoints.....	38
1.8 Non-CADI sync watchpoints.....	38
1.8.1 syncLevel definitions.....	39
1.8.2 Controlling and observing the syncLevel.....	40
1.9 SCADI.....	42
1.9.1 Intended uses of CADI and SCADI.....	42
1.9.2 Responsibilities of the SCADI caller.....	43
1.9.3 SCADI interface access.....	43
1.9.4 SCADI semantics.....	45
1.10 User mode networking.....	46

1.11 TAP/TUN networking.....	48
1.11.1 TAP/TUN networking limitations.....	48
1.11.2 Setting up a network connection for Microsoft Windows.....	48
1.11.3 Configuring the networking environment for Microsoft Windows.....	49
1.11.4 Commands for tap_setup_32.exe or tap_setup_64.exe for Microsoft Windows.....	49
1.11.5 Uninstalling networking for Microsoft Windows.....	50
1.11.6 Setting up a network connection for Red Hat Enterprise Linux.....	50
1.11.7 Setting up a network connection for Ubuntu Linux.....	52
1.11.8 Configuring the networking environment for Linux.....	54
1.11.9 Solutions to networking issues on Linux.....	55
1.11.10 Disabling and re-enabling networking for Linux.....	55
1.11.11 Uninstalling networking for Linux.....	56
1.12 Using parameters to set port values.....	56
1.13 PVBUS C++ transaction and Tx_Result classes.....	57
1.13.1 Class pv::TransactionGenerator.....	57
1.13.2 TransactionGenerator efficiency considerations.....	58
1.13.3 Enum pv::AccessWidth.....	58
1.13.4 Class pv::Transaction.....	58
1.13.5 Class pv::ReadTransaction.....	59
1.13.6 Class pv::WriteTransaction.....	59
1.14 Visualisation library.....	60
1.14.1 LISA visualisation models.....	61
1.14.2 Visualisation library C++ classes.....	61
2. Protocols.....	66
2.1 AMBA-PV protocols.....	66
2.1.1 AMBAPV protocol.....	66
2.1.2 AMBAPVACE protocol.....	69
2.1.3 AMBAPVSignal protocol.....	72
2.1.4 AMBAPVSignalState protocol.....	72
2.1.5 AMBAPVValue protocol.....	72
2.1.6 AMBAPVValue64 protocol.....	72
2.1.7 AMBAPVValueState protocol.....	73
2.1.8 AMBAPVValueState64 protocol.....	73
2.2 Clocking protocols.....	73
2.2.1 ClockRateControl protocol.....	73

2.2.2 TimerCallback protocol.....	74
2.2.3 TimerCallback64 protocol.....	74
2.2.4 TimerControl protocol.....	74
2.2.5 TimerControl64 protocol.....	75
2.3 Debug interface protocols.....	75
2.3.1 CADIDisassemblerProtocol protocol.....	76
2.3.2 CADIProtocol protocol.....	77
2.4 Peripheral protocols.....	79
2.4.1 AudioControl protocol.....	79
2.4.2 CharacterLCD protocol.....	80
2.4.3 FlashLoaderPort protocol.....	80
2.4.4 GUIPollCallback protocol.....	80
2.4.5 ICS307Configuration protocol.....	80
2.4.6 KeyboardStatus protocol.....	81
2.4.7 LCD protocol.....	81
2.4.8 LCDLayoutInfo protocol.....	82
2.4.9 MMC_Protocol protocol.....	82
2.4.10 MouseStatus protocol.....	83
2.4.11 PL080_DMACE_DmaPortProtocol protocol.....	83
2.4.12 PS2Data protocol.....	84
2.4.13 PVBUSSlaveControl protocol.....	84
2.4.14 PVDevice protocol.....	87
2.4.15 PVTransactionMaster protocol.....	87
2.4.16 SerialData protocol.....	88
2.4.17 SMMUv3AEMIdentifyProtocol protocol.....	89
2.4.18 TZFilterControl protocol.....	89
2.4.19 VirtualEthernet protocol.....	90
2.5 Power management protocols.....	90
2.5.1 PChannel protocol.....	90
2.6 Processor protocols.....	91
2.6.1 CoprocBusProtocol protocol.....	91
2.6.2 CounterInterface protocol.....	97
2.6.3 GICv3Comms protocol.....	97
2.6.4 InstructionCount protocol.....	98
2.6.5 v8EmbeddedCrossTrigger_controlprotocol protocol.....	99
2.7 Signaling protocols.....	99

2.7.1 Signal protocol.....	99
2.7.2 StateSignal protocol.....	99
2.7.3 Value protocol.....	100
2.7.4 Value_64 protocol.....	100
2.7.5 ValueState protocol.....	100
3. Fast Models components.....	101
3.1 Component differences.....	101
3.2 Bridge components.....	103
3.2.1 AMBAPV2PVBus.....	104
3.2.2 AMBAPVACE2PVBus.....	105
3.2.3 AMBAPVSignal2SGSignal.....	107
3.2.4 AMBAPVSignalState2SGStateSignal.....	107
3.2.5 AMBAPVValue2SGValue.....	108
3.2.6 AMBAPVValue2SGValue64.....	108
3.2.7 AMBAPVValue642SMMUv3AEMIdentify.....	109
3.2.8 AMBAPVValue642VECB.....	109
3.2.9 AMBAPVValueState2SGValueState.....	110
3.2.10 AMBAPVValueState2SGValueState64.....	110
3.2.11 BroadcastSignal2AMBAPVSignal.....	111
3.2.12 Clock2SystemC.....	112
3.2.13 ClockRateConversion.....	112
3.2.14 ClockSignal2SC_ClockSignal.....	115
3.2.15 CoprocBus2SystemC.....	115
3.2.16 CounterInterface2SystemC.....	116
3.2.17 InstructionCount2SystemC.....	116
3.2.18 LCD2SystemC.....	117
3.2.19 PChannel2SystemC.....	118
3.2.20 PVBus2AMBAPV.....	118
3.2.21 PVBus2AMBAPVACE.....	121
3.2.22 PVBusBridge.....	124
3.2.23 SC_ClockSignal2ClockSignal.....	125
3.2.24 SGSignal2AMBAPVSignal.....	126
3.2.25 SGStateSignal2AMBAPVSignalState.....	126
3.2.26 SGValue2AMBAPVValue.....	127
3.2.27 SGValue2AMBAPVValue64.....	127

3.2.28 SGValueState2AMBAPVValueState.....	128
3.2.29 SGValueState2AMBAPVValueState64.....	129
3.2.30 SMMUv3AEMIdentify2AMBAPVValue64.....	129
3.2.31 SystemC2Clock.....	130
3.2.32 SystemC2CoprocBus.....	130
3.2.33 SystemC2CounterInterface.....	131
3.2.34 SystemC2InstructionCount.....	131
3.2.35 SystemC2LCD.....	132
3.2.36 SystemC2PChannel.....	133
3.2.37 SystemC2VirtualEthernet.....	133
3.2.38 SystemC2v7VGICConfig.....	134
3.2.39 VECB2AMBAPVValue64.....	134
3.2.40 VirtualEthernet2SystemC.....	135
3.2.41 v7VGICConfig2SystemC.....	135
3.3 Bus components.....	136
3.3.1 Labeller.....	137
3.3.2 LabellerForDMA330.....	139
3.3.3 LabellerForDMA330_cpp.....	141
3.3.4 LabellerForGPUProtMode.....	142
3.3.5 LabellerForGPUProtMode_cpp.....	144
3.3.6 Labeller_cpp.....	145
3.3.7 MSIRewriter.....	146
3.3.8 PASSwitch.....	148
3.3.9 PVBUS4KBTo1KBSplitter.....	150
3.3.10 PVBUSCache.....	151
3.3.11 PVBUSDecoder.....	152
3.3.12 PVBUSExclusiveMonitor.....	153
3.3.13 PVBUSExclusiveSquasher.....	156
3.3.14 PVBUSExclusiveSquasher_cpp.....	156
3.3.15 PVBUSLogger.....	157
3.3.16 PVBUSMapper.....	158
3.3.17 PVBUSMaster.....	159
3.3.18 PVBUSModifier.....	161
3.3.19 PVBUSRouter.....	162
3.3.20 PVBUSSlave.....	162
3.3.21 PVMemoryProtectionEngine.....	165

3.3.22 PVWriteBuffer.....	168
3.3.23 SimplePVBusMaster.....	169
3.3.24 TZSwitch.....	170
3.4 Clocking components.....	172
3.4.1 ClockDivider.....	172
3.4.2 ClockGate.....	174
3.4.3 ClockSelector.....	175
3.4.4 ClockTimer.....	181
3.4.5 ClockTimer64.....	181
3.4.6 ClockTimerThread.....	182
3.4.7 ClockTimerThread64.....	183
3.4.8 ClockTimerThread64_cpp.....	184
3.4.9 ClockTimerThread_cpp.....	184
3.4.10 MasterClock.....	185
3.4.11 PLLControl.....	186
3.4.12 ScalableClockControl.....	187
3.4.13 SwitchedClockControl.....	197
3.5 Core components.....	205
3.5.1 AEMv8RMPCT.....	206
3.5.2 AEMvACT.....	351
3.5.3 ARMAEMv8MCT.....	596
3.5.4 ARMCortexA5CT.....	868
3.5.5 ARMCortexA5MPx1CT.....	881
3.5.6 ARMCortexA7x1CT.....	896
3.5.7 ARMCortexA8CT.....	914
3.5.8 ARMCortexA9MPx1CT.....	928
3.5.9 ARMCortexA9UPCT.....	943
3.5.10 ARMCortexA15x1CT.....	956
3.5.11 ARMCortexA17x1CT.....	974
3.5.12 ARMCortexA32x1CT.....	992
3.5.13 ARMCortexA34x1CT.....	1012
3.5.14 ARMCortexA35x1CT.....	1033
3.5.15 ARMCortexA53x1CT.....	1053
3.5.16 ARMCortexA55CT.....	1075
3.5.17 ARMCortexA55CT_CortexA75CT.....	1099
3.5.18 ARMCortexA55CT_CortexA76CT.....	1136

3.5.19 ARMCortexA55CT_CortexA78CT.....	1174
3.5.20 ARMCortexA57x1CT.....	1211
3.5.21 ARMCortexA65AECT.....	1232
3.5.22 ARMCortexA65AECT_CortexA76AECT.....	1258
3.5.23 ARMCortexA65CT.....	1298
3.5.24 ARMCortexA72x1CT.....	1322
3.5.25 ARMCortexA73x1CT.....	1342
3.5.26 ARMCortexA75CT.....	1362
3.5.27 ARMCortexA76AECT.....	1386
3.5.28 ARMCortexA76CT.....	1411
3.5.29 ARMCortexA77CT.....	1435
3.5.30 ARMCortexA78AECT.....	1459
3.5.31 ARMCortexA78CCT.....	1484
3.5.32 ARMCortexA78CT.....	1509
3.5.33 ARMCortexA510CT.....	1533
3.5.34 ARMCortexA510CT_CortexA710CT.....	1565
3.5.35 ARMCortexA510CT_CortexA710CT_CortexX2CT.....	1615
3.5.36 ARMCortexA510CT_CortexA710CT_CortexX3CT.....	1682
3.5.37 ARMCortexA510CT_CortexA715CT_CortexX3CT.....	1753
3.5.38 ARMCortexA520AECT.....	1825
3.5.39 ARMCortexA520CT.....	1859
3.5.40 ARMCortexA520CT_CortexA720CT.....	1892
3.5.41 ARMCortexA520CT_CortexA720CT_CortexX4CT.....	1944
3.5.42 ARMCortexA520CT_CortexA725CT.....	2016
3.5.43 ARMCortexA520CT_CortexA725CT_CortexX925CT.....	2069
3.5.44 ARMCortexA710CT.....	2141
3.5.45 ARMCortexA715CT.....	2171
3.5.46 ARMCortexA720AECT.....	2204
3.5.47 ARMCortexA720CT.....	2237
3.5.48 ARMCortexA725CT.....	2269
3.5.49 ARMCortexA725CT_CortexX925CT.....	2300
3.5.50 ARMCortexM0CT.....	2352
3.5.51 ARMCortexM0PlusCT.....	2358
3.5.52 ARMCortexM3CT.....	2365
3.5.53 ARMCortexM4CT.....	2372
3.5.54 ARMCortexM7CT.....	2380

3.5.55 ARMCortexM23CT.....	2391
3.5.56 ARMCortexM33CT.....	2409
3.5.57 ARMCortexM35PCT.....	2430
3.5.58 ARMCortexM52CT.....	2451
3.5.59 ARMCortexM55CT.....	2473
3.5.60 ARMCortexM85CT.....	2495
3.5.61 ARMCortexR4CT.....	2517
3.5.62 ARMCortexR5x1CT.....	2529
3.5.63 ARMCortexR7x1CT.....	2542
3.5.64 ARMCortexR8x1CT.....	2556
3.5.65 ARMCortexR52PlusCT.....	2572
3.5.66 ARMCortexR52x1CT.....	2590
3.5.67 ARMCortexR82AECT.....	2608
3.5.68 ARMCortexR82CT.....	2637
3.5.69 ARMCortexX1CCT.....	2666
3.5.70 ARMCortexX1CT.....	2690
3.5.71 ARMCortexX2CT.....	2715
3.5.72 ARMCortexX3CT.....	2743
3.5.73 ARMCortexX4CT.....	2777
3.5.74 ARMCortexX925CT.....	2808
3.5.75 ARMNeoverseE1CT.....	2839
3.5.76 ARMNeoverseN1CT.....	2863
3.5.77 ARMNeoverseN2CT.....	2887
3.5.78 ARMNeoverseN3CT.....	2916
3.5.79 ARMNeoverseV1CT.....	2945
3.5.80 ARMNeoverseV2CT.....	2970
3.5.81 ARMNeoverseV3AECT.....	3001
3.5.82 ARMNeoverseV3CT.....	3033
3.5.83 ARMSC000CT.....	3066
3.5.84 ARMSC300CT.....	3073
3.6 Media components.....	3079
3.6.1 D71.....	3079
3.6.2 DP500.....	3086
3.6.3 DP500x2.....	3089
3.6.4 DP550.....	3091
3.6.5 DP550x2.....	3094

3.6.6 DP650.....	3098
3.6.7 DP650x2.....	3101
3.6.8 Mali_C5x_streaming_sink.....	3104
3.6.9 Mali_C7x_streaming_sink.....	3105
3.6.10 Mali_C55.....	3107
3.6.11 Mali_C71.....	3112
3.6.12 Mali_C78.....	3114
3.6.13 Mali_Cxx_streaming_camera.....	3115
3.6.14 Mali_G51.....	3117
3.6.15 Mali_G71.....	3119
3.6.16 Mali_G72.....	3121
3.6.17 Mali_G76.....	3124
3.6.18 Mali_G78AE.....	3126
3.6.19 Mali_G710.....	3129
3.6.20 Mali_G715.....	3132
3.6.21 Mali_G720.....	3135
3.6.22 Mali_G725.....	3137
3.6.23 Mali_T624.....	3139
3.6.24 V61.....	3141
3.6.25 V550.....	3145
3.7 Peripheral components.....	3149
3.7.1 AudioOut_File.....	3149
3.7.2 AudioOut_SDL.....	3150
3.7.3 Base_PowerController.....	3151
3.7.4 CMSDK_Watchdog_cpp.....	3154
3.7.5 Cluster_Temperature_Sensor.....	3155
3.7.6 CombinedMessagingUnit.....	3157
3.7.7 DebugAccessPort.....	3168
3.7.8 DebugROM.....	3170
3.7.9 DualClusterSystemConfigurationBlock.....	3184
3.7.10 DummyAPB.....	3188
3.7.11 ElfLoader.....	3191
3.7.12 FlashLoader.....	3193
3.7.13 GICv3CommsLogger.....	3194
3.7.14 GICv3CommsPVBUS.....	3195
3.7.15 GICv3ProtocolChecker.....	3196

3.7.16 GUIPoll.....	3197
3.7.17 HostBridge.....	3198
3.7.18 HostSerialInterface.....	3200
3.7.19 Infra6_Core_Mgr_Clock_Ctrl.....	3201
3.7.20 Infra6_DVFS_Manager.....	3202
3.7.21 Infra6_LCP_Extended_Control.....	3204
3.7.22 Infra6_LCP_External_Control.....	3206
3.7.23 Infra6_MSCP_PIK.....	3207
3.7.24 Infra6_MSCP_RAS_n_INIT_Ctrl.....	3210
3.7.25 Infra6_SRAM_ECC_RAS.....	3211
3.7.26 Infra6_System_PIK.....	3212
3.7.27 Infra6p6_AP_Mgmt_Regs.....	3218
3.7.28 Infra6p6_LCP_Extended_Ctrl_Regs.....	3220
3.7.29 Infra6p6_LCP_External_Ctrl_Regs.....	3221
3.7.30 IntelStrataFlashJ3.....	3223
3.7.31 Interrupt_Router.....	3225
3.7.32 IoTSS3_ManagerSecurityController.....	3227
3.7.33 IoTSS3_MemoryProtectionController.....	3229
3.7.34 IoTSS3_SecureAccessConfig.....	3231
3.7.35 Juno_sysregs.....	3235
3.7.36 LS64TestingFIFO.....	3236
3.7.37 MemoryElement_TZC400_cpp.....	3237
3.7.38 MemoryMappedGenericTimer.....	3239
3.7.39 MemoryMappedGenericWatchdog.....	3241
3.7.40 NonVolatileCounter.....	3242
3.7.41 PCIeATC.....	3244
3.7.42 PLLClockControl.....	3246
3.7.43 PPUMultiThreadModeSwitch.....	3247
3.7.44 PS2Keyboard.....	3247
3.7.45 PS2Mouse.....	3248
3.7.46 PVBUSGICv3Comms.....	3249
3.7.47 PVMetaDataController.....	3250
3.7.48 PchannelListener.....	3252
3.7.49 RAMDevice.....	3253
3.7.50 RAMDevice_cpp.....	3255
3.7.51 ROM.....	3257

3.7.52 RSE_Integ_Regs.....	3258
3.7.53 RandomNumberGenerator.....	3259
3.7.54 RealTimeLimiter.....	3260
3.7.55 RealtimeClockTimer.....	3262
3.7.56 RemapDecoder.....	3262
3.7.57 RootKeyStorage.....	3263
3.7.58 SI_System_Ctrl_Regs.....	3266
3.7.59 SMCF.....	3268
3.7.60 SMMUv3TestEngine.....	3276
3.7.61 STLBusGasket.....	3278
3.7.62 SerialCrossover.....	3279
3.7.63 SoC_SOR.....	3279
3.7.64 SystemIdUnit.....	3281
3.7.65 System_Time_Synchronization_Control_cpp.....	3283
3.7.66 TRNG.....	3284
3.7.67 TelnetTerminal.....	3285
3.7.68 Temperature.....	3289
3.7.69 UnusedPrimeCell.....	3290
3.7.70 VirtioBlockDevice.....	3291
3.7.71 VirtioBlockDeviceMMIO.....	3293
3.7.72 VirtioNetMMIO.....	3295
3.7.73 VirtioP9Device.....	3300
3.7.74 VirtioPCIBlockDevice.....	3302
3.7.75 VirtioRNG.....	3304
3.7.76 VirtualEthernetCrossover.....	3308
3.7.77 VirtualEthernetHub3.....	3308
3.7.78 VisEventRecorder.....	3309
3.7.79 Visualisation_sdl2.....	3312
3.7.80 WarningMemory.....	3324
3.7.81 v8EmbeddedCrossTrigger_Matrix.....	3325
3.8 Scheduler components.....	3326
3.8.1 AsyncSignal.....	3326
3.8.2 SchedulerInterface.....	3327
3.8.3 SchedulerThread.....	3327
3.8.4 SchedulerThreadEvent.....	3328
3.8.5 SchedulerThreadEvent_cpp.....	3328

3.8.6 SchedulerThread_cpp.....	3329
3.9 Signals components.....	3329
3.9.1 AndGate.....	3329
3.9.2 FrequencyProbe.....	3330
3.9.3 LabellerMasterIdExtendedIdUserFlag.....	3331
3.9.4 LabellerUserSignals.....	3333
3.9.5 OrGate.....	3335
3.9.6 SGSignalBuffer.....	3335
3.9.7 SignalDriver.....	3336
3.9.8 SignalInverter.....	3337
3.9.9 SignalLogger.....	3337
3.9.10 Value64Logger.....	3338
3.9.11 ValueLogger.....	3339
3.9.12 WideAndGate.....	3339
3.9.13 WideOrGate.....	3340
3.10 SystemIP components.....	3341
3.10.1 AHCI_SATA.....	3341
3.10.2 AddressTranslationUnit.....	3343
3.10.3 BP141_TZMA.....	3345
3.10.4 BP147_TZPC.....	3347
3.10.5 CCI400.....	3348
3.10.6 CCI500.....	3353
3.10.7 CCI550.....	3362
3.10.8 CCN502.....	3372
3.10.9 CCN504.....	3380
3.10.10 CCN508.....	3389
3.10.11 CCN512.....	3397
3.10.12 CI700.....	3406
3.10.13 CMN600.....	3444
3.10.14 CMN600AE.....	3457
3.10.15 CMN600CMLHub.....	3470
3.10.16 CMN650.....	3472
3.10.17 CMN650R2.....	3486
3.10.18 CMN700.....	3500
3.10.19 DMA350.....	3545
3.10.20 DMC500.....	3557

3.10.21 DMC520.....	3559
3.10.22 DMC620.....	3561
3.10.23 DMC_400.....	3563
3.10.24 EthosU55.....	3566
3.10.25 EthosU65.....	3568
3.10.26 EthosU85.....	3570
3.10.27 Firewall.....	3573
3.10.28 GIC500.....	3584
3.10.29 GIC500_ClusterPorts.....	3613
3.10.30 GIC500_Filter.....	3643
3.10.31 GIC600.....	3672
3.10.32 GIC600AE.....	3689
3.10.33 GIC600AE_Filter.....	3705
3.10.34 GIC600_Filter.....	3722
3.10.35 GIC625.....	3738
3.10.36 GIC625_Filter.....	3745
3.10.37 GIC700.....	3753
3.10.38 GIC700_Filter.....	3783
3.10.39 GIC720AE.....	3813
3.10.40 GIC720AE_Filter.....	3843
3.10.41 GIC_400.....	3872
3.10.42 GIC_IRI.....	3881
3.10.43 GIC_IRI_Filter.....	3939
3.10.44 ICS307.....	3997
3.10.45 IDAU.....	4000
3.10.46 IntegrityChecker.....	4258
3.10.47 IoTSS3_SystemControl.....	4260
3.10.48 IoTSS_AccessControlGate.....	4264
3.10.49 IoTSS_PeripheralProtectionController.....	4265
3.10.50 KeyManagementUnit.....	4267
3.10.51 LifeCycleManager.....	4272
3.10.52 MMC.....	4277
3.10.53 MMU_400.....	4284
3.10.54 MMU_400_BASE.....	4315
3.10.55 MMU_500.....	4320
3.10.56 MMU_500_BASE.....	4356

3.10.57 MMU_600.....	4366
3.10.58 MMU_700.....	4393
3.10.59 MMU_S3.....	4425
3.10.60 MemoryMappedCounterModule.....	4463
3.10.61 MessageHandlingUnit.....	4467
3.10.62 MessageHandlingUnitV2.....	4473
3.10.63 MessageHandlingUnitV3.....	4476
3.10.64 NI700.....	4481
3.10.65 PL011_Uart.....	4486
3.10.66 PL011_Uart_cpp.....	4491
3.10.67 PL022_SSP.....	4495
3.10.68 PL030_RTC.....	4497
3.10.69 PL031_RTC.....	4498
3.10.70 PL041_AACI.....	4500
3.10.71 PL050_KMI.....	4501
3.10.72 PL061_GPIO.....	4503
3.10.73 PL080_DMAC.....	4504
3.10.74 PL110_CLCD.....	4507
3.10.75 PL111_CLCD.....	4509
3.10.76 PL180_MCI.....	4510
3.10.77 PL192_VIC.....	4512
3.10.78 PL310_L2CC.....	4515
3.10.79 PL330_DMAC.....	4521
3.10.80 PL340_DMC.....	4533
3.10.81 PL350_SMC.....	4543
3.10.82 PL350_SMC_NAND_FLASH.....	4566
3.10.83 PL370_HDLCD.....	4569
3.10.84 PL390_GIC.....	4571
3.10.85 PPUMTWakerequest.....	4582
3.10.86 PPUv0.....	4583
3.10.87 PPUv1.....	4587
3.10.88 PPUv1_Cluster_Wakerequest_Logic.....	4595
3.10.89 RSE_CPU_Private_Region.....	4596
3.10.90 SMMUv3AEM.....	4597
3.10.91 SMSC_91C111.....	4658
3.10.92 SP804_Timer.....	4660

3.10.93 SP805_Watchdog.....	4662
3.10.94 SP810_SysCtrl.....	4663
3.10.95 SSU.....	4667
3.10.96 SecureAlarmManager.....	4668
3.10.97 SecurelCache.....	4670
3.10.98 SystemFMU.....	4673
3.10.99 System_RAS_Agent.....	4674
3.10.100 TZC_400.....	4676
3.10.101 TZFilterUnit.....	4690
3.10.102 TZIC.....	4690
3.10.103 TrustedRAM.....	4691
3.10.104 VHT_VIOBridge.....	4693
3.10.105 VHT_VSIBridge.....	4694
3.10.106 VHT_VSocket.....	4695
3.10.107 v7_VGIC.....	4695
4. Plug-ins for Fast Models.....	4700
4.1 Loading a plug-in.....	4700
4.1.1 --plugin command-line option.....	4700
4.1.2 scx::scx_load_plugin() method.....	4700
4.1.3 FM_TRACE_PLUGINS environment variable.....	4701
4.2 Customizing a plug-in.....	4702
4.3 ArchMsgTrace.....	4702
4.3.1 ArchMsgTrace - parameters.....	4704
4.4 ASTFplugin.....	4704
4.4.1 ASTFplugin - parameters.....	4705
4.4.2 ASTFplugin usage notes.....	4706
4.4.3 Additional ASTF support in Fast Models.....	4706
4.4.4 ASTF tools.....	4707
4.5 BranchPrediction.....	4707
4.5.1 BranchPrediction - parameters.....	4708
4.5.2 BranchPrediction output example.....	4708
4.5.3 Other ways to report branch mispredictions.....	4709
4.6 CADIIPCRemoteConnection.....	4709
4.6.1 CADIIPCRemoteConnection - parameters.....	4710
4.7 CDE.....	4710

4.7.1 CDETester.....	4711
4.7.2 CDETester - parameters.....	4712
4.7.3 CDEConstant.....	4713
4.7.4 CDEConstant - parameters.....	4713
4.7.5 Implementing a CDE plug-in.....	4714
4.7.6 CDE API.....	4716
4.8 CDELoader.....	4724
4.8.1 CDELoader - parameters.....	4725
4.8.2 ACI library implementation.....	4726
4.8.3 ACIConstant example ACI library.....	4727
4.8.4 Custom instruction mnemonics.....	4727
4.8.5 ACI library API.....	4727
4.9 Crypto.....	4742
4.9.1 Crypto - parameters.....	4744
4.10 GDBRemoteConnection.....	4745
4.10.1 GDBRemoteConnection limitations.....	4745
4.11 GenericCounter.....	4746
4.11.1 GenericCounter - parameters.....	4746
4.12 GenericTrace.....	4747
4.12.1 GenericTrace - parameters.....	4748
4.12.2 GenericTrace plug-in usage example.....	4749
4.12.3 Mapping between SYSREG_UPDATE trace sources and SPSR registers.....	4751
4.13 libete-plugin.....	4752
4.13.1 libete-plugin - parameters.....	4752
4.14 ListTraceSources.....	4754
4.14.1 ListTraceSources - parameters.....	4754
4.15 PipelineModel.....	4755
4.15.1 CortexA53PipelineModel - parameters.....	4756
4.15.2 InOrderPipelineModel - parameters.....	4757
4.15.3 PipelineModel example.....	4757
4.15.4 Naming the plug-in instance.....	4758
4.15.5 Example command lines.....	4758
4.15.6 PipelineModel output.....	4760
4.16 RunTimeParameterTest.....	4761
4.17 Sidechannel.....	4761
4.18 TarmacText.....	4761

4.18.1 TarmacText - parameters.....	4762
4.19 TarmacTrace.....	4763
4.19.1 TarmacTrace - parameters.....	4763
4.19.2 TarmacTrace file format.....	4765
4.19.3 Tarmac Trace output example.....	4767
4.19.4 Instruction trace.....	4769
4.19.5 Program flow trace.....	4770
4.19.6 Register trace.....	4771
4.19.7 Cache maintenance trace.....	4772
4.19.8 Cache content trace.....	4773
4.19.9 Translation table walk trace.....	4774
4.19.10 Granule protection table walk trace.....	4776
4.19.11 TLB trace.....	4777
4.19.12 Event trace.....	4778
4.19.13 Processor memory access trace.....	4780
4.19.14 Processor memory update trace.....	4781
4.19.15 Memory bus trace.....	4782
4.20 ToggleMTIPlugin.....	4785
4.20.1 ToggleMTIPlugin - parameters.....	4785
4.20.2 How to use ToggleMTIPlugin.....	4785
5. Fast Models examples.....	4787
5.1 CADI examples.....	4787
5.2 LISA examples.....	4788
5.3 Build and run an FVP example.....	4789
5.4 LISAPlus examples.....	4793
5.5 MTI examples.....	4793
5.6 SystemCExport examples.....	4794
5.7 Build and run an EVS platform example.....	4795
6. Base Platform.....	4797
6.1 Base Platform RevC.....	4798
6.1.1 Base Platform RevC block diagram.....	4799
6.1.2 BasePlatformPCIRevC component.....	4799
6.1.3 Base Platform RevC PCIeRootComplex.....	4802
6.1.4 Base Platform RevC SMMUv3AEM.....	4803
6.1.5 Base Platform RevC PCIe device assignment.....	4804

6.1.6 Base Platform RevC legacy PCI interrupts.....	4805
6.1.7 Base Platform RevC MSI-X.....	4805
6.2 Base Platform startup configuration for v9 cores.....	4805
6.3 Base Platform memory.....	4806
6.3.1 Base Platform memory map.....	4806
6.3.2 Base Platform secure memory.....	4809
6.3.3 BaseR Platform memory map.....	4809
6.3.4 Base Platform DRAM.....	4810
6.4 Base Platform interrupt assignments.....	4810
6.5 Base Platform clocks.....	4812
6.6 Base Platform parameters.....	4813
6.7 Base Platform components.....	4814
6.7.1 Base Platform Base_PowerController component.....	4814
6.7.2 Base Platform DebugAccessPort component.....	4819
6.7.3 Simulator visualization - parameters.....	4820
6.7.4 VE_SysRegs component.....	4821
6.8 Base Platform VE compatibility.....	4822
6.8.1 Base Platform GICv2.....	4822
6.8.2 Base Platform GICv3.....	4823
6.8.3 Base Platform system global counter.....	4823
6.8.4 Disable security in the Base Platform.....	4823
6.9 Base Platform unsupported VE features.....	4824
6.9.1 Base Platform memory aliasing at 0x08_00000000.....	4824
6.9.2 Boot ROM alias at 0x00_0800_0000 in the Base Platform.....	4824
6.9.3 Changed parameters in the Base Platform.....	4824
7. Microcontroller Prototyping System 2.....	4825
7.1 MPS2.....	4825
7.2 MPS2 platform types.....	4825
7.3 MPS2 memory maps.....	4826
7.3.1 MPS2 memory map for models without the Armv8-M additions.....	4826
7.3.2 MPS2 memory map for models with the Armv8-M additions.....	4827
7.4 MPS2 interrupt assignments.....	4833
7.5 MPS2 - differences between models and hardware.....	4833
8. Versatile Express Model.....	4836
8.1 VE.....	4836

8.2 VE memory map for Cortex-A series.....	4838
8.3 VE memory map for Cortex-R series.....	4840
8.4 VE interrupt assignments for Cortex-A series.....	4841
8.5 VE interrupt assignments for Cortex-R series.....	4842
8.6 VE parameters.....	4843
8.6.1 VE instantiation parameters.....	4843
8.6.2 VE secure memory parameters.....	4844
8.6.3 VE switch S6.....	4844
8.7 VE components.....	4845
8.7.1 VEVisualisation component.....	4845
8.8 VE_SysRegs component.....	4849
8.8.1 VE_SysRegs.....	4849
8.8.2 VE_SysRegs ports.....	4849
8.8.3 VE_SysRegs parameters.....	4850
8.8.4 VE_SysRegs registers.....	4850
8.8.5 VE_SysRegs - verification and testing.....	4851
8.9 Differences between the VE hardware and the system models.....	4851
8.9.1 Memory map.....	4851
8.9.2 Memory aliasing.....	4852
8.9.3 VE hardware features absent.....	4852
8.9.4 VE hardware features different.....	4852
Proprietary notice.....	4853
Product and document information.....	4855
Product status.....	4855
Revision history.....	4855
Conventions.....	4857
Useful resources.....	4859

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. The timing is arranged so that on average the processor executes one instruction per clock cycle.



Timing annotation can cause the cycle count and instruction count to differ. For more details, see [Timing Annotation](#) in the Fast Models User Guide.

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.

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 types of failure-case, but not all types.

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.

The Cortex-A9 and Cortex-A5 models do not model device-accurate MESI behavior. The Cortex-A15 and Cortex-A7 models do simulate hardware MOESI state handling, and can handle cache-to-cache snoops. In addition, they model the AMBA 4 ACE cache-coherency protocols over their PVBUS ports, so can be connected to a model of an ACE Coherent Interconnect (such as the CCI-400 model) to simulate coherent sharing of cache contents between processors.

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	-

PMUv1 event	MTI trace sources	Notes
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.
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. Note: 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	-

PMUv2 event	MTI source	Notes
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.

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.



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.



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.

1.5.9 CADI interactions with processor behavior

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

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

M series

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

A and R series

Asserting the reset pin causes the processor to stop executing instructions and the PC to be given its reset value, which depends on VINITHI, and the SP is UNKNOWN. See the Reset section in the [ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition](#). On deasserting the reset pin, the processor begins fetching and executing instructions.

The behavior of reset on M series processors interacts differently with CADI debugging functionality than it does for A and R series processors.

On A and R series processors, after the reset pin of the processor is asserted, a new application can be loaded using the `CADIExecLoadApplication()` call in CADI. This loads an application into memory, and sets the PC if a start address is specified in the application. When reset is deasserted, the processor begins fetching and executing from the start address as expected. Similarly, if a debugger asserts reset on the processor, it can modify memory with a sequence of `CADIMemWrite()` calls, update the PC with a `CADIRegWrite()` call, and when reset is deasserted the processor begins to fetch and execute from the PC.

On M series processors, these techniques do not work because after reset is deasserted the processor updates SP_main and the PC, overwriting the settings made by the CADI calls.

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

1.6 Fast Models CADI implementation

Fast Models implement a subset of CADI functionality.

The following CADI methods are not supported by Fast Models:

CADI class:

Register API

`CADIGetCommittedPCs()`

Memory API

`CADIMemGetOverlays()`

Virtual Memory API

`PhysicalToVirtual()`

Cache API

- `CADIGetCacheInfo()`
- `CADICacheRead()`
- `CADICacheWrite()`

Execution API

- `CADIExecLoadApplication()`. Implemented by CPU components only
- `CADIExecUnloadApplication()`
- `CADIExecGetLoadedApplications()`. Implemented by CPU components only. It only returns the most recently loaded application if more than one have been loaded.
- `CADIExecAssertException()`
- `CADIExecGetPipeStages()`
- `CADIExecGetPipeStageFields()`
- `CADIGetCycleCount()`

CADICallbackObj class:

- `appliOpen()`
- `appliClose()`
- `cycleTick()`
- `killInterface()`

CADIDisassembler class:

- `GetSourceReferenceForAddress()`
- `GetAddressForSourceReference()`
- `GetInstructionType()`

CADIDisassemblerCB class:

- `ReceiveSourceReference()`

The following tables describe the implementation of other CADI functionality by processor models:

Table 1-4: CADI broker implementation

Feature	Implemented	Remarks
Factories	Yes	Model DLL provides a single factory.
Instances	Yes	Factory can only instantiate one model at a time.

Table 1-5: CADI target implementation

Feature	Implemented	Remarks
# breakpoints	Yes	No intrinsic limit.
Breakpoint disabling	Yes	You can disable global breakpoints, but not user breakpoints set through <code>CADIBptSet()</code> . Debuggers usually disable breakpoints with <code>CADIBptClear()</code> and re-enable them with <code>CADIBptSet()</code> , so this does not affect debugger GUIs, but it does affect the CADI C++ interface. Global breakpoints are built-in and permanent, so <code>CADIBptSet()/Clear()</code> cannot set or delete them. With <code>CADIBptConfigure()</code> , you can only enable/disable these global breakpoints to stop on certain processor events.
Breakpoint ignore count	No	-
Breakpoint formal conditions	No	-
Breakpoint free-form conditions	No	-
Breakpoint setting	Yes, but not permitted while target is running.	-
Breakpoint types	Yes: <code>CADI_BPT_MEMORY</code> , <code>CADI_BPT_PROGRAM</code> , <code>CADI_BPT_PROGRAM_RANGE</code>	All registers in the Vector register group have disabled register breakpoints. You cannot create new register breakpoints.
Breakpoint triggers	Yes: <code>CADI_BPT_TRIGGER_ON_MODIFY</code> , <code>CADI_BPT_TRIGGER_ON_READ</code> , <code>CADI_BPT_TRIGGER_ON_WRITE</code>	<code>CADI_BPT_TRIGGER_ON_WRITE</code> approximates <code>CADI_BPT_TRIGGER_ON_MODIFY</code> . If either is set, the simulation stops when a value is written, even when it does not change.
<code>CADIExecSingleStep()</code>	Yes	Fast Models ignores the <code>stepCycle</code> and <code>stepOver</code> arguments.
Intrusive debug	Yes	Stop/start can affect processor scheduling: single stepping and multiple stepping both change scheduling, and so are intrusive.
Memory read/write	Yes, but not permitted while target is running.	-
Reset	Yes	One reset level is supported. When calling <code>CADIExecReset()</code> , Arm recommends the simulation should use a <code>syncLevel</code> of 1 or higher for best results.
Runnable	Yes	-
Stop simulation	Yes	-

1.7 CADI sync watchpoints

The CADI/Synchronous CADI (SCADI) interfaces of processor components support watchpoints. Memory components like RAMDevice do not.

When the simulator hits a watchpoint, the CADI/SCADI interface emits a sequence, typically only one, of `modeChange (CADI_EXECMODE_Bpt, bptNumber)` callbacks, where `bptNumber` is the breakpoint ID of the watchpoint. After this sequence it sends a `modeChange (CADI_EXECMODE_stop)` callback. Only after the debugger receives this `stop` callback might it inspect the state of the model.

You can read additional information about the last of the hit watchpoints from these CADI registers declared by the processor in register group `simulation`:

memoryBptPC

The PC of the instruction that caused the watchpoint to hit.

memoryBptAccessVA

Virtual address of the access or sub-access that caused the watchpoint to hit.

memoryBptAccessSize

Size in bytes of the access or sub-access that caused the watchpoint to hit.

memoryBptAccessRW

Type of access or sub-access that caused the watchpoint to hit: 1 = read, 2 = write, 3 = both, 0 = no access.

These registers are not memory (or CPnn-) mapped anywhere and are not accessible to target programs. These registers are read-only. `memoryBptPC` and `memoryBptAccessVA` are 32 or 64 bit depending on the architecture. `memoryBptAccessSize` and `memoryBptAccessRW` are 32 bit.

When multiple accesses have hit watchpoints at the same time, for example during the same instruction, the information contained in these registers is valid and consistent only for one of these accesses.

Whenever at least one CADI watchpoint is set on a processor, the `syncLevel` on that processor is at least 2.

1.8 Non-CADI sync watchpoints

This section describes `syncLevel` enum values, semantics, and behaviors, with `syncLevel` values listed from fast to slow simulation, from fewer to more requirements.

1.8.1 syncLevel definitions

Definitions for the possible syncLevel values.

syncLevel 0: OFF

- 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 will be 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 CADI debugging while no CADI watchpoint is set.



When using CADI reset on a model, for best results, Arm recommends using a syncLevel of 1 or higher.

syncLevel 1: SYNC_STATE

- The simulation runs slightly slower than syncLevel 0. SCADI 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/memory from within a peripheral or memory access.

syncLevel 2: POST_INSN_IO

- As for syncLevel 1, except that you can stop the simulation synchronously from within all LD/ST and similar instructions. The simulation stops immediately after the current LD/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/ST instructions.
- Use cases: CADI watchpoints, external breakpoints, stopping from within LD/ST-related MTI callbacks.

syncLevel 3: POST_INSN_ALL

- 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 will be 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.

1.8.2 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.

1.9 SCADI

A Synchronous CADI (SCADI) interface enables you to request a synchronous stop of the simulation. It offers direct synchronous, but unsynchronized, register and memory read access.

The intention is that this interface is used by code that is inherently synchronized with the simulation, in particular from code that is part of the simulation itself (simulation thread). Examples of code that could use this interface are SystemC and LISA peripheral device or bus access functions, and MTI callback functions that are always called synchronously by the simulation itself.



The SCADI interface is not a replacement for CADI. It is an additional interface with clear semantics that differ slightly from CADI.

1.9.1 Intended uses of CADI and SCADI

Component Architecture Debug Interface (CADI) and Synchronous CADI (SCADI) have separate design principles.

CADI

Used by debuggers and simulation controllers to control the simulation. Execution control functions are always asynchronous and are usually called from a non-simulation thread, usually the debugger GUI thread. You can also use CADI to read/write registers, memory and so on, but only from non-simulation threads, for example only from the debugger thread.

SCADI

Implements a specific subset of CADI functions, that is, mainly read/write registers and memory, set and clear breakpoints, and get disassembly. You can only use SCADI from the simulation thread itself and from threads that can make sure on their own that the simulation is currently blocked, for example a debugger thread while it is sure that the simulation is in a stable state. SCADI is intended to be used from within peripheral read/write accesses while the simulation is running, or from within MTI callbacks that the simulation is running.

SCADI does not implement and execute control functions except `CADIExecStop()`, because only stop makes sense from within the simulation thread. Asynchronous functions such as `CADIExecContinue()` and `CADIExecSingleStep()` are not supported by SCADI, so for these you must use CADI instead.

The `SCADI::CADIExecStop()` function is the exception, because for CADI this means “stop when it is convenient for the simulation”, which is asynchronous, but `CADIExecStop()` is synchronous and means “stop now”, which you enable with the appropriate `syncLevel >= 2`.

The guidelines are:

- Use CADI for execution control.
- Use CADI or SCADI for read/write registers and memory, depending on the situation:

- Use SCADI if the caller can make sure that the accesses are (potentially inherently) synchronized with the simulation.
- Use CADI if called from a debugger thread that does not know exactly whether the simulation is running or is in a stable state.

1.9.2 Responsibilities of the SCADI caller

Synchronous interface means that the caller of SCADI functions is always responsible for ensuring that the simulation is in a stable state.

Being in a stable state means that the simulation does not advance while the call is being made. The caller is also responsible for meeting all host thread and synchronization requirements of any external bus slaves that are directly or indirectly connected to the memory bus.

1.9.3 SCADI interface access

You can obtain the SCADI interface from the same `CAInterface` pointers that provide the CADI and MTI interfaces.

This is done using the `CAInterface::ObtainInterface()` method. The interface names for the normal CADI and MTI interfaces are `CADI::IFNAME()` and `ComponentTraceInterface::IFNAME()`, respectively.

The interface name for the SCADI interface to pass into `ObtainInterface()` is `"eslapi.SCADI2"` and the interface revision is 0. There are no `IFNAME()` and `IFREVISION()` static functions defined for SCADI, and a plain string constant and integer constant (0) are used instead.

The pointer returned by `CAInterface::ObtainInterface("eslapi.SCADI2")` must be cast into an `eslapi::CADI` class pointer, that is, the same class as for `eslapi.CADI2`, the normal CADI interface class.

1.9.3.1 Access to SCADI from within LISA components

To access the SCADI interface of the LISA component itself, use the `getSCADI()` function. This returns the SCADI interface of the LISA component itself.

To access the SCADI interfaces of other components in the system, use the `getGlobalInterface()` function. This returns a `CAInterface` pointer to a simulation-wide Component Registry.

You then have to use this registry pointer with the `obtainComponentInterfacePointer()` function to obtain the `SystemTraceInterface`. This interface provides a list of all components in the system that provide trace data. You can use this list to access the SCADI interfaces of all the components.

1.9.3.2 Access to SCADI from within SystemC

To access the SCADI interfaces from within SystemC, use the `scx_get_global_interface()` function. This returns a `CAInterface` pointer to a simulation-wide Component Registry.

You then have to use this registry pointer with the `obtainComponentInterfacePointer()` function to obtain the `SystemTraceInterface`. This interface provides a list of all components in the system that provide trace data. You can use this list to access the SCADI interfaces of all the components.

1.9.3.3 Access to SCADI from MTI plug-ins

MTI plug-ins can gain access to SCADI interfaces by using the `CAInterface` pointer, which is passed as a parameter to the `RegisterSimulation()` function.

This pointer has the same semantics as the `CAInterface` pointer returned by the `getGlobalInterface()` functions in the SystemC or LISA use cases.

1.9.3.4 Example code for retrieving a SCADI interface pointer of a component

This code demonstrates how to use `getGlobalInterface()` to retrieve a SCADI interface pointer of a specific component.

From within LISA, use `getGlobalInterface()`. From within SystemC, it is assumed that you have a pointer to the Fast Models platform called `platform`. From within an MTI plug-in, you do not have to call `getGlobalInterface()`, and you can use the `CAInterface` pointer passed to `RegisterSimulation()`.

```
#include "sg/SGComponentRegistry.h"
eslapi::CADI *Peripheral::get_SCADI_interface(const char *instanceName)
{
    // get access to MTI interface (SystemTraceInterface)
    // - this code is for a SystemC peripheral
    // - for LISA this would be just 'getGlobalInterface()'
    // - for MTI plugins this would be just using the CAInterface *caif pointer passed
    // into RegisterSimulation()
    eslapi::CAInterface *globalInterface = scx::scx_get_global_interface();
    const char *errorMessage = "(no error)";
    MTI::SystemTraceInterface *mti = 0;
    if (globalInterface)
        mti = sg::obtainComponentInterfacePointer<MTI::SystemTraceInterface>
            (globalInterface,
             "mtiRegistry",
             &errorMessage);

    if (mti == 0)
    {
        printf("ERROR:MTI interface not found! (%s)\n", errorMessage);
        return 0;
    }
    // find component with instanceName
    eslapi::CAInterface *compif = 0;
    for (MTI::SystemTraceInterface::TraceComponentIndex i = 0;
         i < mti->GetNumOfTraceComponents();
         i++)
    {
        const char *name = mti->GetComponentTracePath(i);
        if (verbose)
```



```

        printf("TEST MTI component '%s'\n", name);
        if (strcmp(name, instanceName) == 0)
            compif = mti->GetComponentTrace(i);
    }
    if (!compif)
    {
        printf("ERROR:instance '%s' not found while trying to get SCADI! \n",
               instanceName);
        return 0;
    }
    // get and return SCADI interface
    return static_cast<eslapi::CADI *>(compif->ObtainInterface("eslapi.SCADI2", 0,
0));
}

```

1.9.4 SCADI semantics

This section lists the available subset of functionality and the differences in semantics from the CADI interface.

The SCADI interface provides a subset of functionality of the CADI interface. All functions in SCADI differ from the CADI functions, that is, the caller is responsible for satisfying the constraints and requirements in terms of synchronization with the simulation.

These functions have the same semantics as in CADI:

- CADIRegGetGroups()
- CADIRegGetMap()
- CADIRegGetCompount()
- CADIMemGetSpaces()
- CADIMemGetBlocks()
- CADIMemGetOverlays()
- CADIGetParameters()
- CADIGetParameterInfo()
- CADIXfaceGetFeatures()

The following debug read access functions have the same semantics as in CADI, but the values that these read access functions return might differ from the values returned by the corresponding CADI functions because some of the accessed resource values might change during the execution of an instruction.

The values of resources that are modified by the current instruction are **UNKNOWN**, except the PC, which always reflects the address of the last instruction that was issued. Reading and writing registers and memory while the simulation is running is generally only useful for `syncLevel >= SL_SYNC_STATE`. For `SL_OFF`, all registers and all memory locations are always **UNKNOWN** while the simulation is running.

- CADIRegRead()
- CADIMemRead()

- `CADIGetParameterValues()`
- `CADIGetInstructionCount()`
- `CADIGetPC()`
- `CADIGetDisassembler()`

The following write access functions always provide write access to the register in the `simulation` register group. They also might provide limited write access semantics to certain resources. A SCADI implementation might choose not to support any write access to core registers and memory:

- `CADIRegWrite()`
- `CADIMemWrite()`
- `CADISetParameters()`

The breakpoint functions have nearly all the same semantics as in the CADI interface, except that changes to breakpoints only become effective at the next stop event checkpoint defined by the current `syncLevel`. Changes might also only become visible in `CADIBptRead()` and `CADIBptGetList()` at this next synchronization point.

- `CADIBptGetList()`
- `CADIBptRead()`
- `CADIBptSet()`
- `CADIBptClear()`
- `CADIBptConfigure()`

Execution control is limited to stopping the simulation. The simulation stops at the next synchronization point as defined by the current `syncLevel`:

- `CADIExecStop()`.

All other functions are unsupported, and return `CADI_STATUS_CmdNotSupported`.

1.10 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 CADI parameters to activate the components:

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](#)

1.11 TAP/TUN networking

This section describes TAP/TUN networking.

1.11.1 TAP/TUN networking limitations

This section describes general limitations of TAP/TUN networking for models.

DHCP

If the host uses Dynamic DNS, it inserts records into DNS. If you manage this host with DHCP, then 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.

WiFi

Most WiFi adaptors do not implement the required support for TAP networking to work.

1.11.2 Setting up a network connection for Microsoft Windows

This section describes how to set up a network connection.

Before you begin

The Fast Models networking environment needs the Third Party IP add-on for the Fast Model Portfolio.

Procedure

1. Close all non-essential applications.
2. Install the TAP driver and configure it:
 - a) Locate the Fast Models Portfolio, by default `C:\Program Files\ARM\FastModelsPortfolio_X.Y\`.
 - b) Execute the `ModelNetworking\add_adapter_64.bat` file.
3. Select **Start > Control Panel > Network Connections** and locate the newly installed TAP device.
4. Select at least one real Ethernet adapter connection. Press the **Ctrl** key to multi-select.
5. Right-click and select **Bridge Connections**.

1.11.3 Configuring the networking environment for Microsoft Windows

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

About this task



- This procedure has not been tested with wireless network adapters.
- Firewall software might block network traffic in the network bridge, and might result in a networking failure in the model. If the model does not work after configuration, check the firewall settings.

Procedure

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

SMSC_91C111:

```
hostbridge.interfaceName="ARM<x>"
smc_91c111.enabled=1
```

VirtioNetMMIO:

```
virtio_net.hostbridge.interfaceName="ARM<x>"
virtio_net.enabled=1
```

`ARM<x>` is an adapter that is built into the network bridge, and is 0 by default. Use `x+1` if `ARM<x>` exists.

2. Set another parameter on the device, if you have to enable promiscuous mode for the TAP device.
 - a. Select **Start > Run > cmd.exe**.
 - b. Enter the following commands.

```
netsh bridge show adapter
netsh bridge set adapter <N> forcecompatmode=enable
```

`<N>` is the id number of the TAP adapter that the first command lists.

1.11.4 Commands for `tap_setup_32.exe` or `tap_setup_64.exe` for Microsoft Windows

This section lists the commands for use with `tap_setup_32.exe` or `tap_setup_64.exe`. These files are located in the `ModelNetworking` directory.

help

Print help information.

set_media

Configure the TAP devices to `Always Connected`.

set_perm

Configure the TAP devices to `Non-admin Accessible`.

restart

Restart the TAP devices.

list_dev

List available TAP devices and output to a file.

rename

Rename the device to **ARMx**.

install <infile> <id>

Install a TAP Win32 or Win64 adapter.

remove <dev>

Remove TAP Win32 or Win64 adapters. Set <dev> to `All` to remove all tap devices.

setup <infile> <id>

Automated setup process.

1.11.5 Uninstalling networking for Microsoft Windows

This section describes how to uninstall TAP adapters from the base location of the Fast Models Portfolio, by default `C:\Program Files\ARM\FastModelsPortfolio_X.Y`.

Procedure

1. Run the `remove_all_adapters_32.bat` Or `remove_all_adapters_64.bat` file. These files are located in the `ModelNetworking` directory.
2. If the uninstallation does not delete the bridge, delete it manually:
 - a) Close all non-essential applications.
 - b) Select **Network Connections**.
 - c) Right-click on the bridge and select **Delete**.

1.11.6 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 in the installation package, but we recommend that you use the standard Linux bridge utilities, which are in the Linux distribution.

About this task



- 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.
-



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
```

1.11.7 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.



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
```



armbr0 must be explicitly started.

This step disconnects and reconnects all network interfaces.

1.11.8 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



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](#)

1.11.9 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`.



Refer to the vendor's documentation manual.

1.11.10 Disabling and re-enabling networking for Linux

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

About this task



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
```

1.11.11 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.

1.12 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.

1.13 PVBUS C++ transaction and Tx_Result classes

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

1.13.1 Class `pv::TransactionGenerator`

This class provides efficient mechanisms for bus masters to generate transactions that are transmitted over the `pvbuss_m` 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);
};
```

1.13.2 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`.

1.13.3 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`

1.13.4 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();

};

```

1.13.5 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();
};

```

1.13.6 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();
};
```

1.14 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 81

[LISA visualisation models](#) on page 61

[Visualisation library C++ classes](#) on page 61

1.14.1 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](#) on page 4836

1.14.2 Visualisation library C++ classes

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

1.14.2.1 C++ classes inclusion

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

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

1.14.2.2 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.

1.14.2.3 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.

1.14.2.4 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.

1.14.2.5 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.

1.14.2.6 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]
```

1.14.2.7 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 AMBA-PV protocols

The AMBA-PV components and protocols permit you to model a platform that interfaces with an Arm® AMBA®-based system.

The protocols are:

AMBAPV

Models the AMBA® protocols AXI4, AXI3, AHB and APB.

AMBAPVACE

Models the AMBA® ACE and DVM protocols.

AMBAPVSignal

Models interrupts.

AMBAPVSignalState

Transfers and receives signal states.

AMBAPVValue

Models propagation of 32-bit integer values between components.

AMBAPVValue64

Models propagation of 64-bit integer values between components.

AMBAPVValueState

Permits a master to retrieve the current value from a slave, using 32-bit integer values.

AMBAPVValueState64

Permits a master to retrieve the current value from a slave, using 64-bit integer values.

There are ready-to-use components that provide you with conversions between protocols.

Related information

[AMBA-PV Extensions to TLM User Guide](#)

2.1.1 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.

The LISA definition for the AMBAPV protocol is `$PVLIB_HOME/LISA/AMBAPVProtocol.lisa`. The behaviors of the protocol are:

read()

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.

```
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
```

write()

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.

```
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
```

debug_read()

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_read(int socket_id, const sc_dt::uint64 &addr,
           unsigned char *data, unsigned int length,
           const amba_pv::amba_pv_control *ctrl):
    unsigned int
```

debug_write()

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.

```
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
```

b_transport()

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.

```
b_transport(int socket_id, amba_pv::amba_pv_transaction &trans, sc_core::sc_time
&t)
```

transport_dbg()

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.

```
transport_dbg(int socket_id, amba_pv::amba_pv_transaction &trans,
    sc_core::sc_time &t):
    unsigned int
```

get_direct_mem_ptr()

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.

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.

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

atomic_store() and atomic_load()

These optional slave behaviors complete an AtomicStore or AtomicLoad transaction with the specified data. The data is used by the atomic transaction in the specified endianness.

The only difference between the two behaviors is that AtomicStore does not return data, while AtomicLoad returns the original data.

```
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

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
```


atomic_swap()

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

```
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
```

atomic_compare()

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.

It always returns the original data at the target address.

```
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
```

invalidate_direct_mem_ptr()

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.

```
invalidate_direct_mem_ptr(int socket_id, sc_dt::uint64 start_range,
                        sc_dt::uint64 end_range)
```

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.

2.1.2 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 behaviors of the protocol are:

b_transport()

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.

```
b_transport(int socket_id, amba_pv::amba_pv_transaction &trans, sc_core::sctime
&t)
```

transport_dbg()

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.

```
transport_dbg(int socket_id, amba_pv::amba_pv_transaction &trans,
sc_core::sctime &t): unsigned int
```

get_direct_mem_ptr()

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.

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

b_snoop()

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_snoop(int socket_id, amba_pv::amba_pv_transaction &trans, sc_core::sctime &t)
```

snoop_dbg()

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.

```
snoop_dbg(int socket_id, amba_pv::amba_pv_transaction &trans, sc_core::sctime
&t): unsigned int
```

invalidate_direct_mem_ptr()

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.

```
invalidate_direct_mem_ptr(int socket_id,  sc_dt::uint64 start_range,  
                           sc_dt::uint64 end_range)
```

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.

2.1.3 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.

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.

2.1.4 AMBAPVSignalState protocol

This protocol defines two behaviors that permit a master to change the state of signals such as interrupt 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.

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.

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.

2.1.5 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.

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.

2.1.6 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.

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.

2.1.7 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.

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.

get_state(unsigned 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.

2.1.8 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.

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.

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.

2.2 Clocking protocols

The clocking components and protocols provide a mechanism for systems to regulate the execution rate of components.

All clocking components that communicate use an opaque `ClockSignal` protocol. `ClockSignal` protocols have no behaviors that the user can invoke.

Related information

[Clocking components](#) on page 172

2.2.1 ClockRateControl protocol

This protocol sets the ratio of this component.

- `set(uint32_t mul, uint32_t div) : void`
- `set64(uint64_t mul, uint64_t div) : void`

Set the multiplier and divider that determine the clock divider ratio:

```
clk_outfreq= clk_infreq * mul / div
```

2.2.2 TimerCallback protocol

This protocol invokes a behavior on the component that set the timer. Do not use the `timer_control` port of the timer during the invoked behavior.

signal(): uint32_t

Invoked by the timer when the timer countdown expires. The invoked behavior returns the number of ticks after which it will be called again or 0 to make the timer one-shot.

2.2.3 TimerCallback64 protocol

This protocol invokes a behavior on the component that set the timer. Do not use the `timer_control` port of the timer during the invoked behavior.

signal(): uint64_t

Invoked by the timer when the timer countdown expires. The invoked behavior returns the number of ticks after which it will be called again or 0 to make the timer one-shot.

2.2.4 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.

set(uint32_t ticks): void

Set the timer to count down the given number of ticks.

cancel()

Cancel an active timer, preventing the callback being invoked.

isSet() : bool

Check whether a timer is set to generate a callback.

remaining() : uint32_t

Return how many ticks remain before the callback is invoked.

2.2.5 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.

set(uint64_t ticks): void

Set the timer to count down the given number of ticks.

cancel()

Cancel an active timer, preventing the callback being invoked.

isSet() : bool

Check whether a timer is set to generate a callback.

remaining() : uint64_t

Return how many ticks remain before the callback is invoked.

2.3 Debug interface protocols

LISA components can expose aspects of their internal state so that they become visible and usable in the debugger. Some aspects are supported by the native LISA language, and some are supported by debug interface protocols.

Registers and other state variables

See the LISA+ resource REGISTER declaration, especially the `read_function` and `write_function` attributes.

Memories and other memory-like objects

See the LISA+ resource MEMORY declaration, especially the `read_function` and `write_function` attributes.

Disassembly

See `CADIProtocol` and `CADIDisassemblerProtocol`.

Instruction count

See `CADIProtocol`.

Setting, configuring, and clearing of breakpoints

For example, code breakpoints, register breakpoints, watchpoints. See `CADIProtocol`.

Single stepping and instruction stepping support

See `CADIProtocol`.

If displaying and editing memory and registers are sufficient, you need not implement `CADIProtocol` and `CADIDisassemblerProtocol`. The component must implement them to enable the other debug features.

CADIProtocol and CADIDisassemblerProtocol permit you to implement the features on a Component Architecture Debug Interface (CADI) level, where the component code takes responsibility for the implementation of these interfaces.

Related information

[Component Architecture Debug Interface User Guide](#)

[LISA+ Language for Fast Models Reference Guide](#)

2.3.1 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, that can have any name and only need to be implemented when disassembly must be exposed in the debugger. The functionality of this port is then exposed by CADIProtocol::CADIGetDisassembler(). See CADIProtocol for information on how to use this port and CADIDisassemblerAdapter.

In the function slave behavior GetType(): eslapi::CADIDisassemblerType; components must always return eslapi::CADI_DISASSEMBLER_TYPE_STANDARD.

The function slave behavior GetModeCount(): uint32_t; returns the number of supported disassembler modes, and at least 1 mode must be returned.

The function slave behavior GetModeNames(eslapi::CADIDisassemblerCB *callback_); returns information about all supported modes. 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.

The function slave behavior GetCurrentMode(): uint32_t; returns the most suitable mode of disassembly at the time, based on the current state variables of the component:

For the function:

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

components must return eslapi::CADI_DISASSEMBLER_STATUS_ERROR;

For the function:

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

components must return eslapi::CADI_DISASSEMBLER_STATUS_ERROR;

The following function is the main disassembler function:

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

```
// Query if an instruction is a call instruction
slave behavior GetInstructionType(const eslapi::CADIAddr_t
&address, eslapi::CADIDisassemblerInstructionType &insn_type):
    eslapi::CADIDisassemblerStatus;
```

Components must return `insn_type = eslapi::CADI_DISASSEMBLER_INSTRUCTION_TYPE_NOCALL;` and return `eslapi::CADI_DISASSEMBLER_STATUS_OK;`

Related information

[CADIProtocol protocol](#) on page 77

[Component Architecture Debug Interface User Guide](#)

2.3.2 CADIProtocol protocol

This protocol supports debugging. To add breakpoint support, implement `CADIBpt...()` functions.



Note

All CADIProtocol protocol behaviors, that is, all sets of functionalities, are optional. A component only has to implement the set of functions for the functionality that it intends to support.

You must define an internal slave port of this type, and the name of this port must always be `cadi_port`. In this port, you can implement this functionality:

```
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;
```

You need to implement all of these functions to enable any kind of breakpoint for the component. The component needs to maintain and keep track of all existing breakpoints. For example:

- `CADIBptSet()` and `CADIBptClear()` add and remove breakpoints.
- `CADIBptConfigure()` enable and disable breakpoints.
- `CADIBptGetList()` and `CADIBptRead()` return the current state of the breakpoint list.

The component must also implement `CADIModifyTargetFeatures`. This function permits you to override the automatic default `CADITargetFeatures_t` that System Generator provides for this component just before it is returned to the debugger. Specifically, a component that wants to support any kind of breakpoint must override the `handledBreakpoints` and `nrBreakpointsAvailable` fields of `CADITargetFeatures_t`. For example:

```
targetFeatures->handledBreakpoints = CADI_TARGET_FEATURE_BPT_PROGRAM |
    CADI_TARGET_FEATURE_BPT_REGISTER;
    // code and register breakpoints supported
targetFeatures->nrBreakpointsAvailable = 0x7fffffff;
    // virtually infinite number of breakpoints supported.
```

By default, LISA components do not support any type of breakpoints. By implementing `CADIBpt...` functions, you can add breakpoint support. In addition to implementing the stated functions, the component must call `simBreakpointHit(bptNumber)` and then `simHalt()` when an enabled breakpoint is hit. On a breakpoint hit the component must first call `simBreakpointHit()` for each breakpoint that was hit (one or more, usually just one) and then call `simHalt()` once after all `simBreakpointHit()` calls. The `simHalt()` call must always be the last call in the sequence.

A component that wants to provide disassembly must implement the following `CADIGetDisassembler()` behavior and return a `CADIDisassembler` interface implementation. This automatically follows behind the `CADI::CADIGetDisassembler()` and the `CADI::ObtainInterface("eslapi.CADIDisassembler2")` functions.

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

To do this, instantiate a `CADIDisassemblerAdapter` object in behavior `init` and return its address in this `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
```

```
{
    // ...
}
```

The following function implements the instruction stepping a component. 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.

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

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

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

By implementing any of the following functions, the component can enable the instruction and cycle count display:

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



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

2.4 Peripheral protocols

This section describes the peripheral protocols.

2.4.1 AudioControl protocol

This protocol has get and release audio buffer behaviors.

getPVAudioBuffer

Get an underlying host buffer for audio output.

releasePVAudioBuffer

Release an underlying host buffer.

2.4.2 CharacterLCD protocol

This protocol has the behaviors `setLayoutInfo` and `draw`.

setLayoutInfo

Sets the width and height of the touchscreen.

draw

Sets the character information.

2.4.3 FlashLoaderPort protocol

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

loadFlashFile(FlashLoader *) : uint32

Initiate loading of the flash contents.

saveFlashFile(FlashLoader *) : uint32

Save the flash contents to a file.

2.4.4 GUIPollCallback protocol

This protocol defines one method that signals to the visualization component the end of the update period. You can invoke this callback even when the simulation is stopped.

gui_callback(): void

This is sent by the GUIPoll component, at the configured period.

2.4.5 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.

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.

2.4.6 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.

The protocol behaviors are:

keyDown(ATKeyCode code) : void

Sent when a key on the host keyboard is pressed.

keyUp(ATKeyCode code) : void

Sent when a key on the host keyboard is released.

Related information

[VEVisualisation ports](#) on page 4846

2.4.7 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.

The behaviors are:

lock() : VisRasterLayout *

Lock the raster region of the LCD in preparation for rendering.

unlock()

Unlock the raster region, ready to be updated on the screen.

update(int x, int y, unsigned int w, unsigned int h)

Update the selected rectangular area on screen from the raster buffer.

setPreferredLayout(unsigned int width, unsigned int height, unsigned int depth)

A request from the LCD controller to set the preferred size for the raster region, to match the timing parameters used by the LCD controller.

Related information

[VEVisualisation ports](#) on page 4846

2.4.8 LCDLayoutInfo protocol

This protocol has the behavior `setLayoutInfo`.

setLayoutInfo

Sets the width and height of the touchscreen.

2.4.9 MMC_Protocol 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) and some that must be implemented 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, www.mmca.org.

The protocol has these behaviors:

cmd

Commands are sent from the controller to the card using this behavior, which is implemented by the card model. The MMC command is sent with an optional argument. The card responds as defined by the MMC specification. The controller model checks that the response type matches expectations, and updates its state appropriately. The transaction-level protocol does not model start/stop bits or CRCs on the command/response payload.

For data transfer in the card to controller direction:

Rx

After the host and controller have initiated a read through the command interface, the card calls the `Rx` behavior on the controller to provide the read data. The call provides a pointer and a length. The ARM MMC reference model simulates device read latency by waiting a number of clock cycles prior to calling this behavior. If the controller is unable to accept the data, or wants to force a protocol error, it can return false in response to this behavior.

Rx_rdy

A handshake, used by the controller to inform the card that the controller is ready to receive more data. The Arm MMC reference model does not time out, so waits indefinitely for this handshake in a multiple block data transfer.

For data transfer in the controller to card direction:

Tx

After the host and controller have initiated a write through the command interface, the card calls the `Tx` behavior on the controller. The call provides a pointer to an empty buffer to be

written, and a length. The ARM MMC reference model simulates device write latency by waiting a number of clock cycles prior to each buffer being offered.

Tx_done

The controller calls this behavior on the card when the block has been written. The card model can then commit the data to its persistent storage.

The card model must also implement:

cmd_name

This behavior returns the name of the command issued. A card must implement this behavior, but is free to return an empty string for all requests. Only call this behavior for diagnostic messages.

2.4.10 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.

The protocol behaviors are:

mouseMove(int dx, int dy) : void

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

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.

Related information

[VEVisualisation ports](#) on page 4846

2.4.11 PL080_DMAC_DmaPortProtocol protocol

This protocol provides methods to permit handshaking between peripherals and the DMA controller.

request(uint32 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.

0: PL080_REQ_BURST

Burst transfer request.

1: PL080_REQ_SINGLE

Single transfer request.

2: PL080_REQ_LBURST

Last burst request.

3: PL080_REQ_LSINGLE

Last single request.

response(uint32 response): void

Passes responses from the DMA controller to peripherals. The response is a bitfield with the low two bits defined. The response is transient rather than level-sensitive.

0: PL080_RES_TC

Terminal count response.

1: PL080_RES_CLR

Clear request response.

2.4.12 PS2Data protocol

This protocol is for communication between the KMI and a PS/2-like device.

For efficiency, the interface is a parallel byte interface rather than a serial clock/data interface. The behaviors are:

setClockData(enum ps2clockdata) : 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, or is inhibiting communication.

getData() : uint8

Used by the PS/2 device to get command data from the KMI.

putData(uint8 data): void

Used by the PS/2 device to send device data to the KMI.

2.4.13 PVBusSlaveControl protocol

The PVBusSlaveControl protocol enables you to access and modify the underlying memory that PVBusSlave controls.

- `setFillPattern(uint32_t fill1, uint32_t fill2)`

This sets a two-word alternating fill pattern to be used by uninitialized memory.

- `setAccess(pv::bus_addr_t base, pv::bus_addr_t top, pv::accessType type, pv::accessMode mode)`

This reconfigures handling for a region of memory.

`base` (inclusive value) and `top` (exclusive value) specify the address range to configure.

`type` selects what types of bus access must be reconfigured. It can be one of:

- `pv::ACCESSTYPE_READ`
- `pv::ACCESSTYPE_WRITE`
- `pv::ACCESSTYPE_RW`

`mode` controls what happens when an address is accessed. Legal values for the `pv::accessMode` enumeration are:

`pv::ACCESSMODE_MEMORY`

Act as memory. The PVBUSlave manages the underlying storage to provide memory for the selected address range, which can be ROM or RAM, depending on how you configure it to handle bus write transactions.

`pv::ACCESSMODE_DEVICE`

Act as a device. Requests to the selected address range are routed to the PVBUSlave device port, where the necessary behavior can be implemented by the component.

`pv::ACCESSMODE_ABORT`

Generate bus abort signals for any accesses to the selected address range.

`pv::ACCESSMODE_IGNORE`

Ignore accesses to the selected address range. Bus read requests return 0.

- `getReadStorage(pv::bus_addr_t address, pv::bus_addr_t *limit) : const uint8_t *`
- `getWriteStorage(pv::bus_addr_t address, pv::bus_addr_t *limit) : uint8_t *`

These two methods permit you to access the underlying storage that PVBUSlave allocates to implement a region of memory.

The return value is a pointer to the byte that represents the storage corresponding to the address of `base`.

The `limit` pointer returns the device address for the limit of the accessible memory.

The pointer value returned is not guaranteed to remain valid indefinitely. Bus activities, or other calls to the control port, might invalidate the pointer. For example, if a burst transaction straddles a 4KB boundary, PVBUSlave might reallocate the 4KB regions to be contiguous.

- `provideReadStorage(pv::bus_addr_t device_base, pv::bus_addr_t device_limit, const uint8_t *storage)`
- `provideWriteStorage(pv::bus_addr_t device_base, pv::bus_addr_t device_limit, uint8_t *storage)`
- `provideReadWriteStorage(pv::bus_addr_t device_base, pv::bus_addr_t device_limit, uint8_t *storage)`

These methods enable you to allocate blocks of memory that the PVBUSlave can use to manage regions of RAM/ROM. Only use these methods when you require a high degree of control over memory, such as when you require a device to map specific regions of host memory into the simulation.

The memory region pointed to by `storage` must be large enough to contain `(limit - base)` bytes.

After these calls, PVBUSlave controls access to the underlying memory. The owner must call `getWriteStorage()` before modifying the memory contents and `getReadStorage()` before reading the memory contents.

- `provideReadStorageEx(pv::bus_addr_t device_base, pv::bus_addr_t device_limit, const uint8_t *storage, double latency)`
- `provideWriteStorageEx(pv::bus_addr_t device_base, pv::bus_addr_t device_limit, const uint8_t *storage, double latency)`
- `provideReadWriteStorageEx(pv::bus_addr_t device_base, pv::bus_addr_t device_limit, uint8_t *storage, double read_latency, double write_latency)`

These methods take additional parameters to specify average latencies (in seconds) per byte.

In all other aspects they behave the same as `provideReadStorage()`, `provideWriteStorage()` and `provideReadWriteStorage()`, respectively.

- `slave behavior getRegionIterHandle(): uint32_t;`
- `slave behavior getNextRegionInfo(uint32_t iter_handle, pv::PVBUSlaveRegionInfo *info) : bool;`
- `slave behavior closeRegionIterHandle(uint32_t iter_handle);`

These methods form an iterator-like API that allows a PVBUSlave providing 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 nonzero `iter_handle` to identify it.

The caller can then repeatedly call `getNextRegionInfo()` with `iter_handle`. If it finds a region, the behavior returns `true` and writes to the `info` struct if the pointer is non-null. Access the data in the region using `getReadStorage()` OR `getWriteStorage()`.

The implementation can return regions in any order. They can be of any size or alignment, but must not overlap.

The implementation need not report allocated regions that are filled entirely with the default fill pattern, or allocated regions that contain only the data they had at simulation start.

On reaching the last region, the iterator closes automatically. If the handle is invalid or there are no further regions, the behavior returns `false`.

A caller can close an iterator opened by `getRegionIterHandle()` at any time using `closeRegionIterHandle()`. This deallocates the iterator, and further uses of the handle are invalid.

2.4.14 PVDevice protocol

The PVDevice protocol enables you to implement support for memory-mapped device registers. Call the two methods through the device port on the PVBusSlave to handle bus read/write transactions.

`read(pv::ReadTransaction): pv::Tx_Result`

This method permits a device to handle a bus read transaction.

`write(pv::WriteTransaction): pv::Tx_Result`

This method permits a device to handle a bus write transaction.

The PVDevice protocol uses two behaviors to differentiate between transactions originating from the processor (loads and stores) and transactions originating from an attached debugger:

`slave behavior debugRead(pv::ReadTransactiontx) : pv::Tx_Result`

This method enables the device to handle a debug read transaction.

`slave behavior debugWrite(pv::WriteTransactiontx) : pv::Tx_Result`

This method enables the device to handle a debug write transaction.

The `debugRead` and `debugWrite` behaviors are called for all debug transactions.

For an example component that uses this protocol see `$PVLIB_HOME/examples/LISA/BusComponents/BitLatch.lisa`.

Related information

[PVBus C++ transaction and Tx_Result classes](#) on page 57

2.4.15 PVTransactionMaster protocol

This protocol instantiates a `pv::TransactionGenerator` object from a PVBusMaster.

`createTransactionGenerator() : pv::TransactionGenerator *`

This behavior instantiates a new transaction generator to control the bus master. A caller can allocate as many TransactionGenerators as it wants. It is up to the caller to delete TransactionGenerators when they are no longer required. For example:

```
behavior init() {
    tg = master.createTransactionGenerator();
}

behavior terminate() {
    delete tg;
```

}

2.4.16 SerialData protocol

This protocol is implemented as a parallel interface for efficiency. All communication is driven by the master port.

This protocol has behaviors:

dataTransmit(uint16_t data) : void

Used by the master to send data to the slave.

Table 2-1: Bits for dataTransmit()

Bits	Function
15:8	Reserved
7:0	Transmit data

dataReceive(void) : uint16_t

Used by the master to receive data from the slave.

Table 2-2: 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

signalsSet(uint8_t signal) : void

Used by the master to get the current signal status.

Table 2-3: Bits for signalsSet()

Bits	Function
7	Out1
6	Out2
5	RTS
4	DTR
3:0	Reserved

signalsGet() : uint8_t

Used by the master to get the current signal status.

Table 2-4: Bits for signalsGet()

Bits	Function
7:4	Reserved
3	DCD
2	DSR
1	CTS
0	RI

2.4.17 SMMUv3AEMIdentifyProtocol protocol

This protocol is used by the SMMUv3AEM model to identify the SSD, StreamID, and SubStreamID of an incoming transaction. It is only used if the parameter `howto_identify` is set to "use-identify".

```
identify(unsigned tbu_number_, const pv::TransactionAttributes* attributes_, bool*
out_ssd_ns_, unsigned* out_streamid_, unsigned* out_substreamid_) : void`
```

Architecturally, a transaction comes into the SMMUv3AEM 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.
- StreamID.
- SubStreamID and SubStreamID valid. If you set `*out_substreamid_ = ~0u`, that is interpreted as no SubStreamID because SubStreamIDs are 20 or fewer bits.

How these are transported in the system is SoC-dependent.

The SMMUv3AEM requires that the SoC provides a way of determining this information by implementing the `identify()` function.

2.4.18 TZFilterControl protocol

This protocol controls the communication between filter units and control registers in the APB control block.

```
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. Optional slave behavior.

isEnabled() : bool

Check if the filter unit is enabled or not. The APB control block controls the unit. Slave behavior.

isSecureSlave() : bool

Check if the connected slave is secure or not. Optional slave behavior.

setConfig(bool rd_spec_enable, bool wr_spec_enable, uint32_t action) : void

Pass the configurations to the filter. Optional master behavior.

2.4.19 VirtualEthernet protocol

This protocol has the `sendToSlave` and `sendToMaster` behaviors.

sendToSlave(EthernetFrame *frame)

Send an Ethernet frame to the slave port.

sendToMaster(EthernetFrame *frame)

Send an Ethernet frame to the master port.

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`.

2.5 Power management protocols

This section describes the power management protocols.

2.5.1 PChannel protocol

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

The behaviors of the protocol are:

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 (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.

sg::PChannel::presp_t

This 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.

Usage

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/.

2.6 Processor protocols

This section describes the processor protocols.

2.6.1 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 addCoproprocessor () 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 TestValCoprocessor : 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
    TestValCoprocessor()
        : 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
TestValCoprocessor test_cp;
}

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

2.6.1.1 Coprocessor behaviors

A coprocessor can call the following master behaviors:

addCoprocessor(Coprocessor*, int num) : void

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

removeCoprocessor(Coprocessor*, int num) : void

Unregisters the coprocessor from the CPU.

accessIsPriv(void) : bool

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

accessIsNonSecure(void) : bool

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

2.6.1.2 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, as listed in [CoprocState values](#).

2.6.1.2.1 CDP()

Perform a coprocessor data processing operation.

Prototype

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

Parameters

context

context that was registered with the coprocessor interface.

inst

the coprocessor instruction being executed.

Returns

A `CoprocState` value to indicate the new transaction state of the coprocessor, as listed in [CoprocState values](#).

2.6.1.2.2 MCR()

Perform a move to coprocessor register operation.

Prototype

```
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.

Returns

A `CoprocState` value to indicate the new transaction state of the coprocessor, as listed in [CoprocState values](#).

2.6.1.2.3 MRC()

Perform a move from coprocessor register operation.

Prototype

```
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.

Returns

A `CoprocState` value to indicate the new transaction state of the coprocessor, as listed in [CoprocState values](#).

2.6.1.2.4 LDC()

Perform a load coprocessor register from memory operation.

Prototype

```
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.

Returns

A `CoprocState` value to indicate the new transaction state of the coprocessor, as listed in [CoprocState values](#).

2.6.1.2.5 STC()

Perform a store coprocessor register to memory operation.

Prototype

```
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.

Returns

A `CoprocState` value to indicate the new transaction state of the coprocessor, as listed in [CoprocState values](#).

2.6.1.2.6 MCRR()

Perform a move to two coprocessor registers operation.

Prototype

```
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.

Returns

A `CoprocState` value to indicate the new transaction state of the coprocessor, as listed in [CoprocState values](#).

2.6.1.2.7 MRRC()

Perform a move from two coprocessor registers operation.

Prototype

```
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.

Returns

A `CoprocState` value to indicate the new transaction state of the coprocessor, as listed in [CoprocState values](#).

2.6.1.3 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.

Table 2-5: CoprocState values

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

2.6.2 CounterInterface protocol

This protocol connects the `cntvalueb` port on Generic Timer components to MemoryMappedCounterModule components.

This semi-opaque protocol is exportable across a SystemC interface using a custom bridge.

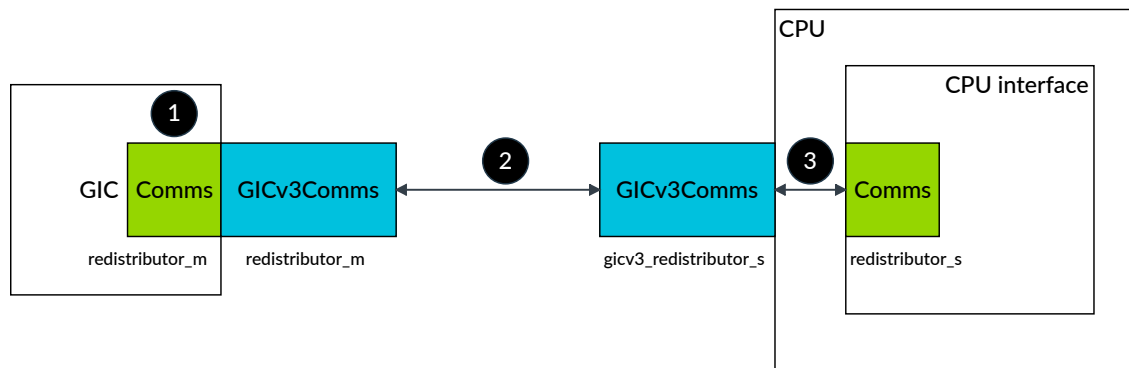
It is a LISA+ protocol.

2.6.3 GICv3Comms protocol

GICv3Comms is a protocol for communication between redistributors in the GIC and the CPU interface. It is supported by the GICv3 and GICv4 models. It is defined in `$PVLIB_HOME/LISA/GICv3Comms.lisa`.

Each GICv3Comms port in the LISA+ file is connected internally to a Comms port, which is a bi-directional communication port. Comms is derived from `sg::port`. The Comms port in the GIC is called `redistributor_m` and the one in the CPU interface is called `redistributor_s`, as shown in the following diagram:

Figure 2-1: GICv3Comms block diagram



Key:

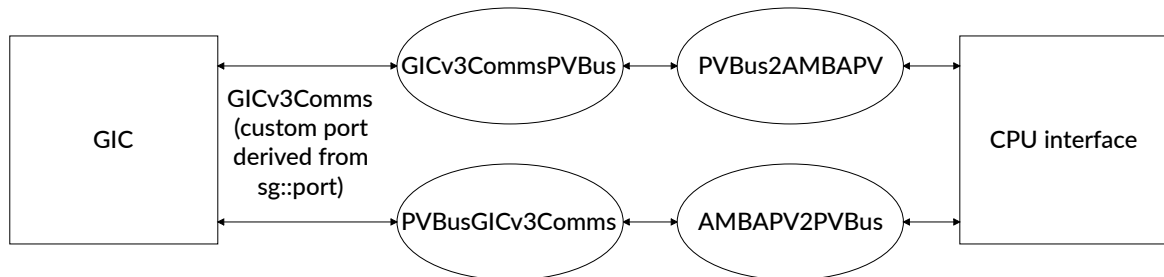
1. GIC contains a Comms port, `redistributor_m`, which the LISA+ file exposes as a GICv3Comms port, `redistributor_m`.
2. `redistributor_m` is connected to the GICv3Comms port in the CPU, `gicv3_redistributor_s`.
3. `gicv3_redistributor_s` forwards the connection to the CPU interface Comms port, `redistributor_s`.

Each redistributor in the GIC communicates through `redistributor_m` to each CPU interface. The number of redistributors in the GIC equals the number of PEs connected.

GIC commands, for example Deactivate, Activate, ActivateAcknowledge, GenerateSGI, Set or Clear interrupt, are communicated through Comms.

The PVBUSGICv3Comms and GICv3CommsPVBUS components convert between the GICv3Comms and PVBUS protocols. The following diagram shows how they can be used in distributed systems:

Figure 2-2: Conversion between GICv3Comms and PVBUS protocols



2.6.4 InstructionCount protocol

This protocol has the behaviors `getValue()` and `getRunState()`.

`getValue()` : `uint64_t`

Obtain the number of instructions executed by the processor.

`getRunState()` : `uint32_t`

Obtain the power/run status of the processor.

Table 2-6: Run state values

Value	State label	Description
0	UNKNOWN	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

2.6.5 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.

2.7 Signaling protocols

Many components use the signaling protocols to indicate changes in state for signals such as interrupt and reset.

Each signaling protocol has two variants:

- One permits components to signal a state change to other components.
- One permits the other components to passively query the current state of the signal.

2.7.1 Signal protocol

The Signal protocol has the setValue behavior.

```
setValue(enum sg::Signal::State) : void  
    Indicates a state change.
```

Valid values for `sg::Signal::State` are:

- `sg::Signal::Set`.
- `sg::Signal::Clear`.

2.7.2 StateSignal protocol

The stateSignal protocol has the setValue and getValue behaviors.

```
setValue(enum sg::Signal::State): void  
    indicates a value change.
```

```
getValue() : sg::Signal::State  
    reads the current value.
```

2.7.3 Value protocol

The Value protocol has the setValue behavior.

setValue(uint32_t value)
indicates a state change.

2.7.4 Value_64 protocol

This protocol provides the rules to communicate with the TrustZone® Memory Adapter (TZMA) to signal the remapped range X.

setValue(uint64_t value)
Sets a 64-bit wide address to the slave port.

2.7.5 ValueState protocol

The ValueState protocol has the setValue and getValue behaviors.

setValue(uint32_t value): void
indicates a value change.

getValue() : uint32_t
reads the current value state.

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 11.26.11 and 11.27.17

Table 3-1: Components added

Component	Quality level
3.7.4 CMSDK_Watchdog_cpp on page 3153	N/A
3.4.2 ClockGate on page 174	N/A
3.4.3 ClockSelector on page 175	N/A
3.4.8 ClockTimerThread64_cpp on page 184	N/A
3.4.9 ClockTimerThread_cpp on page 184	N/A
3.9.2 FrequencyProbe on page 3330	N/A
3.7.19 Infra6_Core_Mgr_Clock_Ctrl on page 3201	N/A
3.7.20 Infra6_DVFS_Manager on page 3202	N/A
3.7.21 Infra6_LCP_Extended_Control on page 3204	N/A
3.7.22 Infra6_LCP_External_Control on page 3205	N/A
3.7.23 Infra6_MSCP_PIK on page 3207	N/A
3.7.24 Infra6_MSCP_RAS_n_INIT_Ctrl on page 3210	N/A
3.7.26 Infra6_System_PIK on page 3212	N/A
3.7.27 Infra6p6_AP_Mgmt_Regs on page 3218	N/A
3.7.28 Infra6p6_LCP_Extended_Ctrl_Regs on page 3220	N/A
3.7.29 Infra6p6_LCP_External_Ctrl_Regs on page 3221	N/A
3.7.32 IoTSS3_ManagerSecurityController on page 3227	N/A
3.7.33 IoTSS3_MemoryProtectionController on page 3229	N/A
3.7.34 IoTSS3_SecureAccessConfig on page 3231	N/A
3.10.47 IoTSS3_SystemControl on page 4260	Alpha support
3.10.48 IoTSS_AccessControlGate on page 4263	Alpha support
3.10.49 IoTSS_PeripheralProtectionController on page 4265	Alpha support
3.7.35 Juno_sysregs on page 3234	N/A
3.3.3 LabellerForDMA330_cpp on page 140	N/A
3.3.5 LabellerForGPUProtMode_cpp on page 143	N/A

Component	Quality level
3.3.6 Labeller_cpp on page 145	N/A
3.7.37 MemoryElement_TZC400_cpp on page 3237	N/A
3.10.66 PL011_Uart_cpp on page 4491	Full support
3.4.11 PLLControl on page 186	N/A
3.3.14 PVBusExclusiveSquasher_cpp on page 156	N/A
3.7.50 RAMDevice_cpp on page 3255	N/A
3.10.89 RSE_CPU_Private_Region on page 4596	Alpha support
3.7.52 RSE_Integ_Regs on page 3258	N/A
3.4.12 ScalableClockControl on page 187	N/A
3.8.5 SchedulerThreadEvent_cpp on page 3328	N/A
3.8.6 SchedulerThread_cpp on page 3329	N/A
3.9.8 SignalInverter on page 3337	N/A
3.7.63 SoC_SOR on page 3279	N/A
3.4.13 SwitchedClockControl on page 197	N/A
3.7.64 SystemIdUnit on page 3281	N/A
3.7.65 System_Time_Synchronization_Control_cpp on page 3283	N/A
3.7.66 TRNG on page 3284	N/A
3.7.68 Temperature on page 3289	N/A
3.7.79 Visualisation_sdl2 on page 3312	N/A
3.9.12 WideAndGate on page 3339	N/A
3.9.13 WideOrGate on page 3340	N/A

Table 3-2: 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	Yes	Yes
ARMCortexA510CT	No	Yes	No	No
ARMCortexR52PlusCT	No	No	No	Yes
ARMCortexR52x1CT	No	No	No	Yes
ARMNeoverseV3AECT	No	No	No	Yes
ARMNeoverseV3CT	No	No	No	Yes
Cluster_Temperature_Sensor	No	No	No	Yes
EthosU85	No	Yes	No	No
GIC600	No	No	No	Yes
GIC600AE	No	No	No	Yes
GIC600AE_Filter	No	No	No	Yes
GIC600_Filter	No	No	No	Yes
GIC625	No	No	No	Yes
GIC625_Filter	No	No	No	Yes
GIC700	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?
GIC700_Filter	No	No	No	Yes
GIC720AE	No	No	No	Yes
GIC720AE_Filter	No	No	No	Yes
MMU_S3	No	Yes	No	No
Mali_C55	No	No	No	Yes
Mali_C71	No	No	No	Yes
Mali_C78	No	No	No	Yes
PPUv1	No	No	No	Yes
PVBus2AMBAPV	No	No	No	Yes
PVBus2AMBAPVACE	No	No	No	Yes
PVBusBridge	No	No	No	Yes
SMCF	No	No	Yes	No
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/SystemCExport/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:

- [2.1 AMBA-PV protocols](#) on page 66

- [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+.

Iris and MTI instances for AMBAPV2PVBUS

This model has the following Iris instances:

Table 3-3: AMBAPV2PVBUS Iris instances

InstanceName	ComponentName
AMBAPV2PVBUS	AMBAPV2PVBUS
AMBAPV2PVBUS.bus_master	PVBUSMaster

This model has the following MTI trace components:

Table 3-4: AMBAPV2PVBUS MTI instances

InstanceName	ComponentName
AMBAPV2PVBUS	AMBAPV2PVBUS
AMBAPV2PVBUS.bus_master	PVBUSMaster

AMBAPV2PVBUS contains the following CADI targets:

- AMBAPV2PVBUS

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.

Ports for AMBAPV2PVBUS

Table 3-5: Ports

Name	Protocol	Type	Description
amba_pv_s	AMBAPV	Slave	-
pvbust_m	PVBUS	Master	-

Parameters for AMBAPV2PVBUS

base_addr

Type

int

Default value

0x0

Base address.

report_errors

Type

bool

Default value

0x0

Report transactions which do not comply with PVBUS protocol requirements.

shareable

Type

bool

Default value

0x1

Shareable default.

3.2.2 AMBAPVACE2PVBUS

AMBA-PV ACE to PVBUS protocol converter. This model is written in LISA+.

Iris and MTI instances for AMBAPVACE2PVBUS

This model has the following Iris instances:

Table 3-6: AMBAPVACE2PVBUS Iris instances

InstanceName	ComponentName
AMBAPVACE2PVBUS	AMBAPVACE2PVBUS

InstanceName	ComponentName
AMBAPVACE2PVBUS.bus_master	PVBusMaster

This model has the following MTI trace components:

Table 3-7: AMBAPVACE2PVBUS MTI instances

InstanceName	ComponentName
AMBAPVACE2PVBUS	AMBAPVACE2PVBUS
AMBAPVACE2PVBUS.bus_master	PVBusMaster

AMBAPVACE2PVBUS contains the following CADI targets:

- AMBAPVACE2PVBUS

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.

Ports for AMBAPVACE2PVBUS

Table 3-8: Ports

Name	Protocol	Type	Description
amba_pv_ace_s	AMBAPVACE	Slave	-
pvbus_m	PVBus	Master	-

Parameters for AMBAPVACE2PVBUS

report_errors

Type

bool

Default value

0x0

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+.

Iris and MTI instances for AMBAPVSignal2SGSignal

This model has the following Iris instances:

Table 3-9: AMBAPVSignal2SGSignal Iris instances

InstanceName	ComponentName
AMBAPVSignal2SGSignal	AMBAPVSignal2SGSignal



Variants of this component also exist with multiple input and output ports.

Ports for AMBAPVSignal2SGSignal

Table 3-10: 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+.

Iris and MTI instances for AMBAPVSignalState2SGStateSignal

This model has the following Iris instances:

Table 3-11: AMBAPVSignalState2SGStateSignal Iris instances

InstanceName	ComponentName
AMBAPVSignalState2SGStateSignal	AMBAPVSignalState2SGStateSignal



Variants of this component also exist with multiple input and output ports.

Ports for AMBAPVSignalState2SGStateSignal

Table 3-12: 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+.

Iris and MTI instances for AMBAPVValue2SGValue

This model has the following Iris instances:

Table 3-13: AMBAPVValue2SGValue Iris instances

InstanceName	ComponentName
AMBAPVValue2SGValue	AMBAPVValue2SGValue



Variants of this component also exist with multiple input and output ports.

Ports for AMBAPVValue2SGValue

Table 3-14: 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+.

Iris and MTI instances for AMBAPVValue2SGValue64

This model has the following Iris instances:

Table 3-15: AMBAPVValue2SGValue64 Iris instances

InstanceName	ComponentName
AMBAPVValue2SGValue64	AMBAPVValue2SGValue64



Variants of this component also exist with multiple input and output ports.

Ports for AMBAPVValue2SGValue64

Table 3-16: 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 3-17: AMBAPVValue642SMMUv3AEMIdentify Iris instances

InstanceName	ComponentName
AMBAPVValue642SMMUv3AEMIdentify	AMBAPVValue642SMMUv3AEMIdentify

Ports for AMBAPVValue642SMMUv3AEMIdentify

Table 3-18: 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 3-19: AMBAPVValue642VECB Iris instances

InstanceName	ComponentName
AMBAPVValue642VECB	AMBAPVValue642VECB

AMBAPVValue642VECB contains the following CADI targets:

- AMBAPVValue642VECB

Ports for AMBAPVValue642VECB

Table 3-20: 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+.

Iris and MTI instances for AMBAPVValueState2SGValueState

This model has the following Iris instances:

Table 3-21: AMBAPVValueState2SGValueState Iris instances

InstanceName	ComponentName
AMBAPVValueState2SGValueState	AMBAPVValueState2SGValueState



Note

Variants of this component also exist with multiple input and output ports.

Ports for AMBAPVValueState2SGValueState

Table 3-22: 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+.

Iris and MTI instances for AMBAPVValueState2SGValueState64

This model has the following Iris instances:

Table 3-23: AMBAPVValueState2SGValueState64 Iris instances

InstanceName	ComponentName
AMBAPVValueState2SGValueState64	AMBAPVValueState2SGValueState64



Variants of this component also exist with multiple input and output ports.

Ports for AMBAPVValueState2SGValueState64

Table 3-24: 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 coverter. This model is written in LISA+.

Iris and MTI instances for BroadcastSignal2AMBAPVSignal

This model has the following Iris instances:

Table 3-25: BroadcastSignal2AMBAPVSignal Iris instances

InstanceName	ComponentName
BroadcastSignal2AMBAPVSignal	BroadcastSignal2AMBAPVSignal

BroadcastSignal2AMBAPVSignal contains the following CADI targets:

- BroadcastSignal2AMBAPVSignal

Ports for BroadcastSignal2AMBAPVSignal

Table 3-26: 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 3-27: Clock2SystemC Iris instances

InstanceName	ComponentName
Clock2SystemC	Clock2SystemC

Clock2SystemC contains the following CADI targets:

- Clock2SystemC

Ports for Clock2SystemC

Table 3-28: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-
current_ticks_s	AMBAPVValueState64	Slave	-
get_clock_s	AMBAPVValueState64	Slave	-
rate_in_Hz_s	AMBAPVValueState64	Slave	-
set_clock_m	AMBAPVValue64	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 3-29: 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 3-30: ClockRateConversion MTI instances

InstanceName	ComponentName
ClockRateConversion.clk_div0	ClockDivider
ClockRateConversion.clk_div1	ClockDivider
ClockRateConversion.clk_div2	ClockDivider
ClockRateConversion.clk_div3	ClockDivider

ClockRateConversion contains the following CADI targets:

- [ClockDivider](#)
- [ClockRateConversion](#)

Ports for ClockRateConversion

Table 3-31: Ports

Name	Protocol	Type	Description
clock	ClockSignal	Slave	-
rate_ctrl[4]	ClockRateControl	Slave	-
rate_hz[4]	Value_64	Master	-

Parameters for ClockRateConversion

clk_div0.div

Type

int

Default value

0x1

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

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

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

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

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

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

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

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 3-32: ClockSignal2SC_ClockSignal Iris instances

InstanceName	ComponentName
ClockSignal2SC_ClockSignal	ClockSignal2SC_ClockSignal

ClockSignal2SC_ClockSignal contains the following CADI targets:

- ClockSignal2SC_ClockSignal

Ports for ClockSignal2SC_ClockSignal

Table 3-33: 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 3-34: CoprocBus2SystemC Iris instances

InstanceName	ComponentName
CoprocBus2SystemC	CoprocBus2SystemC

CoprocBus2SystemC contains the following CADI targets:

- CoprocBus2SystemC

Ports for CoprocBus2SystemC

Table 3-35: 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 3-36: CounterInterface2SystemC Iris instances

InstanceName	ComponentName
CounterInterface2SystemC	CounterInterface2SystemC

CounterInterface2SystemC contains the following CADI targets:

- CounterInterface2SystemC

Ports for CounterInterface2SystemC

Table 3-37: 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	-
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+.

Iris and MTI instances for InstructionCount2SystemC

This model has the following Iris instances:

Table 3-38: InstructionCount2SystemC Iris instances

InstanceName	ComponentName
InstructionCount2SystemC	InstructionCount2SystemC

InstructionCount2SystemC contains the following CADI targets:

- InstructionCount2SystemC



Variants of this component also exist with multiple input and output ports.

Ports for InstructionCount2SystemC

Table 3-39: 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 3-40: LCD2SystemC Iris instances

InstanceName	ComponentName
LCD2SystemC	LCD2SystemC

LCD2SystemC contains the following CADI targets:

- LCD2SystemC

Ports for LCD2SystemC

Table 3-41: 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 3-42: PChannel2SystemC Iris instances

InstanceName	ComponentName
PChannel2SystemC	PChannel2SystemC

PChannel2SystemC contains the following CADI targets:

- PChannel2SystemC

Ports for PChannel2SystemC

Table 3-43: 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+.

Changes in 11.27.17

Parameters removed:

- dmi-container-type

Iris and MTI instances for PVBUS2AMBAPV

This model has the following Iris instances:

Table 3-44: PVBUS2AMBAPV Iris instances

InstanceName	ComponentName
PVBus2AMBAPV	PVBus2AMBAPV
PVBus2AMBAPV.bus_bridge	PVBusBridge

This model has the following MTI trace components:

Table 3-45: PVBUS2AMBAPV MTI instances

InstanceName	ComponentName
PVBus2AMBAPV	PVBus2AMBAPV

InstanceName	ComponentName
PVBus2AMBAPV.bus_bridge	PVBusBridge

PVBus2AMBAPV contains the following CADI targets:

- PVBus2AMBAPV

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.

Ports for PVBUS2AMBAPV

Table 3-46: Ports

Name	Protocol	Type	Description
amba_pv_m	AMBAPV	Master	-
pvbuss	PVBUS	Slave	-

Parameters for PVBUS2AMBAPV

counters-file-name

Type

string

Default value

""

Prefix of the file name to store counters at the end of simulation.

dump-dmi-cache

Type

bool

Default value

0x0

Dumps the content of the DMI cache into a file.

dump-dmi-file-name

Type

string

Default value

""

Prefix of the file name to dump the content of the DMI when requested.

force-dmi-size

Type

bool

Default value

0x1

Force DMI start and end address to be 4kB-aligned.

min-range-to-cache

Type
int

Default value
0x10000

Min DMI range size to cache in the bridge.

size

Type
int

Default value
0x10000000000000

Maximum size of memory region.

3.2.21 PVBUS2AMBAPVACE

PVBus to AMBA-PV ACE protocol converter. This model is written in LISA+.

Changes in 11.27.17

Parameters removed:

- dmi-container-type

Iris and MTI instances for PVBUS2AMBAPVACE

This model has the following Iris instances:

Table 3-47: 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 3-48: PVBUS2AMBAPVACE MTI instances

InstanceName	ComponentName
PVBus2AMBAPVACE	PVBus2AMBAPVACE
PVBus2AMBAPVACE.bus_bridge	PVBusBridge
PVBus2AMBAPVACE.pvbus_tlm_switch	PVBusMapper

PVBus2AMBAPVACE contains the following CADI targets:

- PVBUS2AMBAPVACE

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.

Ports for PVBUS2AMBAPVACE

Table 3-49: 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

""

Prefix of the file name to store counters at the end of simulation.

dmi-cache-name

Type

string

Default value

""

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

Dumps the content of the DMI cache into a file.

dump-dmi-file-name**Type**

string

Default value

""

Prefix of the file name to dump the content of the DMI when requested.

force-dmi-size**Type**

bool

Default value

0x1

Force DMI start and end address to be 4kB-aligned.

min-range-to-cache**Type**

int

Default value

0x10000

Min DMI range size to cache in the bridge.

route-tlm**Type**

bool

Default value

0x1

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

Set bridge mode when connecting to ace-lite ports. If true, the bridge will not deal with SNOOPs.

size**Type**

int

Default value

0x10000000000000

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++.

Changes in 11.27.17

Parameters removed:

- `dmi-container-type`

Iris and MTI instances for PVBusBridge

This model has the following Iris instances:

Table 3-50: PVBusBridge Iris instances

InstanceName	ComponentName
PVBusBridge	PVBusBridge

This model has the following MTI trace components:

Table 3-51: PVBusBridge MTI instances

InstanceName	ComponentName
PVBusBridge	PVBusBridge

Ports for PVBusBridge

Table 3-52: 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
pdbus_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 3-53: SC_ClockSignal2ClockSignal Iris instances

InstanceName	ComponentName
SC_ClockSignal2ClockSignal	SC_ClockSignal2ClockSignal

SC_ClockSignal2ClockSignal contains the following CADI targets:

- SC_ClockSignal2ClockSignal

Ports for SC_ClockSignal2ClockSignal

Table 3-54: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Master	-
sc_clk_in	SC_ClockSignal	Slave	-

3.2.24 SGSignal2AMBAPVSignal

SystemGenerator Signal to AMBA-PV Signal protocol converter. This model is written in LISA+.

Iris and MTI instances for SGSignal2AMBAPVSignal

This model has the following Iris instances:

Table 3-55: SGSignal2AMBAPVSignal Iris instances

InstanceName	ComponentName
SGSignal2AMBAPVSignal	SGSignal2AMBAPVSignal



Variants of this component also exist with multiple input and output ports.

Ports for SGSignal2AMBAPVSignal

Table 3-56: 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+.

Iris and MTI instances for SGStateSignal2AMBAPVSignalState

This model has the following Iris instances:

Table 3-57: SGStateSignal2AMBAPVSignalState Iris instances

InstanceName	ComponentName
SGStateSignal2AMBAPVSignalState	SGStateSignal2AMBAPVSignalState



Variants of this component also exist with multiple input and output ports.

Ports for SGStateSignal2AMBAPVSignalState

Table 3-58: 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+.

Iris and MTI instances for SGValue2AMBAPVValue

This model has the following Iris instances:

Table 3-59: SGValue2AMBAPVValue Iris instances

InstanceName	ComponentName
SGValue2AMBAPVValue	SGValue2AMBAPVValue



Variants of this component also exist with multiple input and output ports.

Ports for SGValue2AMBAPVValue

Table 3-60: 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+.

Iris and MTI instances for SGValue2AMBAPVValue64

This model has the following Iris instances:

Table 3-61: SGValue2AMBAPVValue64 Iris instances

InstanceName	ComponentName
SGValue2AMBAPVValue64	SGValue2AMBAPVValue64



Variants of this component also exist with multiple input and output ports.

Ports for SGValue2AMBAPVValue64

Table 3-62: 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+.

Iris and MTI instances for SGValueState2AMBAPVValueState

This model has the following Iris instances:

Table 3-63: SGValueState2AMBAPVValueState Iris instances

InstanceName	ComponentName
SGValueState2AMBAPVValueState	SGValueState2AMBAPVValueState



Variants of this component also exist with multiple input and output ports.

Ports for SGValueState2AMBAPVValueState

Table 3-64: 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+.

Iris and MTI instances for SGValueState2AMBAPVValueState64

This model has the following Iris instances:

Table 3-65: SGValueState2AMBAPVValueState64 Iris instances

InstanceName	ComponentName
SGValueState2AMBAPVValueState64	SGValueState2AMBAPVValueState64



Variants of this component also exist with multiple input and output ports.

Ports for SGValueState2AMBAPVValueState64

Table 3-66: 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 3-67: SMMUv3AEMIdentify2AMBAPVValue64 Iris instances

InstanceName	ComponentName
SMMUv3AEMIdentify2AMBAPVValue64	SMMUv3AEMIdentify2AMBAPVValue64

Ports for SMMUv3AEMIdentify2AMBAPVValue64

Table 3-68: Ports

Name	Protocol	Type	Description
identify	SMMUv3AEMIdentifyProtocol	Slave	SMMUv3AEMIdentifyProtocol input.
identify_reply	AMBAPVValue64	Slave	From SystemC.
identify_request	AMBAPVValue64	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 3-69: SystemC2Clock Iris instances

InstanceName	ComponentName
SystemC2Clock	SystemC2Clock

SystemC2Clock contains the following CADI targets:

- SystemC2Clock

Ports for SystemC2Clock

Table 3-70: Ports

Name	Protocol	Type	Description
clk_out	ClockSignal	Master	ClockSignal output
current_ticks_m	AMBAPVValueState64	Master	To SystemC.
get_clock_m	AMBAPVValueState64	Master	To SystemC.
rate_in_Hz_m	AMBAPVValueState64	Master	To SystemC.
set_clock_s	AMBAPVValue64	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 3-71: SystemC2CprocBus Iris instances

InstanceName	ComponentName
SystemC2CprocBus	SystemC2CprocBus

SystemC2CprocBus contains the following CADI targets:

- SystemC2CprocBus

Ports for SystemC2CprocBus

Table 3-72: 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 3-73: SystemC2CounterInterface Iris instances

InstanceName	ComponentName
SystemC2CounterInterface	SystemC2CounterInterface

SystemC2CounterInterface contains the following CADI targets:

- SystemC2CounterInterface

Ports for SystemC2CounterInterface

Table 3-74: 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+.

Iris and MTI instances for SystemC2InstructionCount

This model has the following Iris instances:

Table 3-75: SystemC2InstructionCount Iris instances

InstanceName	ComponentName
SystemC2InstructionCount	SystemC2InstructionCount

SystemC2InstructionCount contains the following CADI targets:

- SystemC2InstructionCount



Variants of this component also exist with multiple input and output ports.

Ports for SystemC2InstructionCount

Table 3-76: 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 3-77: SystemC2LCD Iris instances

InstanceName	ComponentName
SystemC2LCD	SystemC2LCD

SystemC2LCD contains the following CADI targets:

- SystemC2LCD

Ports for SystemC2LCD

Table 3-78: 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.
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 3-79: SystemC2PChannel Iris instances

InstanceName	ComponentName
SystemC2PChannel	SystemC2PChannel

SystemC2PChannel contains the following CADI targets:

- SystemC2PChannel

Ports for SystemC2PChannel

Table 3-80: 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 3-81: SystemC2VirtualEthernet Iris instances

InstanceName	ComponentName
SystemC2VirtualEthernet	SystemC2VirtualEthernet

SystemC2VirtualEthernet contains the following CADI targets:

- SystemC2VirtualEthernet

Ports for SystemC2VirtualEthernet

Table 3-82: Ports

Name	Protocol	Type	Description
virtualethernet_m	VirtualEthernet	Master	-
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 3-83: SystemC2v7VGICConfig Iris instances

InstanceName	ComponentName
SystemC2v7VGICConfig	SystemC2v7VGICConfig

SystemC2v7VGICConfig contains the following CADI targets:

- SystemC2v7VGICConfig

Ports for SystemC2v7VGICConfig

Table 3-84: 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 3-85: VECB2AMBAPVValue64 Iris instances

InstanceName	ComponentName
VECB2AMBAPVValue64	VECB2AMBAPVValue64

VECB2AMBAPVValue64 contains the following CADI targets:

- VECB2AMBAPVValue64

Ports for VECB2AMBAPVValue64

Table 3-86: 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 3-87: VirtualEthernet2SystemC Iris instances

InstanceName	ComponentName
VirtualEthernet2SystemC	VirtualEthernet2SystemC

VirtualEthernet2SystemC contains the following CADI targets:

- VirtualEthernet2SystemC

Ports for VirtualEthernet2SystemC

Table 3-88: 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 3-89: v7VGICConfig2SystemC Iris instances

InstanceName	ComponentName
v7VGICConfig2SystemC	v7VGICConfig2SystemC

v7VGICConfig2SystemC contains the following CADI targets:

- v7VGICConfig2SystemC

Ports for v7VGICConfig2SystemC

Table 3-90: 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.
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.

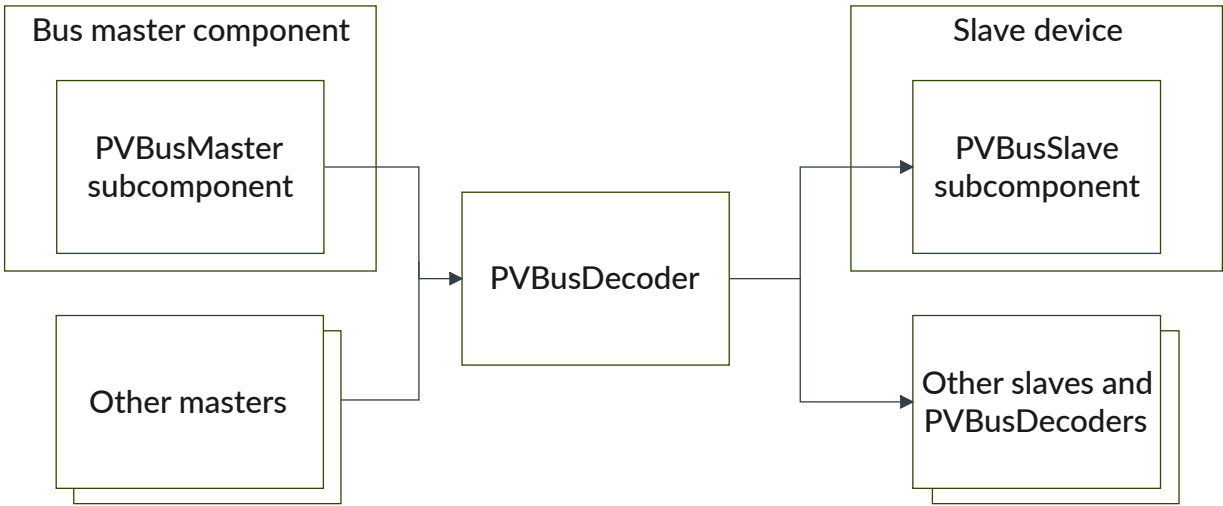
There is no modeling of handshaking or cycle counts. By removing this level of detail, and by using efficient internal communication mechanisms, PVBus components can provide very fast modeling of bus accesses. You must use these components correctly to achieve high simulation speeds.

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+.

Iris and MTI instances for Labeller

This model has the following Iris instances:

Table 3-91: Labeller Iris instances

InstanceName	ComponentName
Labeller	Labeller
Labeller.pvbusmodifier	PVBusMapper

This model has the following MTI trace components:

Table 3-92: Labeller MTI instances

InstanceName	ComponentName
Labeller.pvbusmodifier	PVBusMapper

Labeller contains the following CADI targets:

- Labeller

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.

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.

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.

```
pl370_hdlcd : PL370_HDLCDC();
hdlcd_labeller : Labeller( "label" = 2 );
pl370_hdlcd.pvbus_m => hdlcd_labeller.pvbus_s;
hdlcd_labeller.pvbus_m => output_bus.pvbus_s;
```

Ports for Labeller

Table 3-93: Ports

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Output with modified MasterID.
pvbus_s	PVBus	Slave	Unmodified input.

Parameters for Labeller

label

Type

int

Default value

0x0

The label to apply to all transactions flowing through the labeller.

3.3.2 LabellerForDMA330

This model is written in LISA+.

Iris and MTI instances for LabellerForDMA330

This model has the following Iris instances:

Table 3-94: LabellerForDMA330 Iris instances

InstanceName	ComponentName
LabellerForDMA330	LabellerForDMA330
LabellerForDMA330.pvbusmapper	PVBusMapper

This model has the following MTI trace components:

Table 3-95: LabellerForDMA330 MTI instances

InstanceName	ComponentName
LabellerForDMA330.pvbusmapper	PVBusMapper

LabellerForDMA330 contains the following CADI targets:

- LabellerForDMA330

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.

Ports for LabellerForDMA330

Table 3-96: 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

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

Discriminate between DMA-330 data channels. Channel ID is added to the data label.

dma330_ns_instruction_label**Type**

int

Default value

0x0

The label to apply to all non-secure _instructions_ transactions flowing through the labeller.

dma330_s_instruction_label**Type**

int

Default value

0x0

The label to apply to all secure _instructions_ transactions flowing through the labeller.

3.3.3 LabellerForDMA330_cpp

This model is written in C++.

Iris and MTI instances for LabellerForDMA330_cpp

This model has the following Iris instances:

Table 3-97: LabellerForDMA330_cpp Iris instances

InstanceName	ComponentName
LabellerForDMA330_cpp	LabellerForDMA330
LabellerForDMA330_cpp.PVBusMapper	PVBusMapper

This model has the following MTI trace components:

Table 3-98: LabellerForDMA330_cpp MTI instances

InstanceName	ComponentName
LabellerForDMA330_cpp.PVBusMapper	PVBusMapper

LabellerForDMA330_cpp contains the following CADI targets:

- LabellerForDMA330

Ports for LabellerForDMA330_cpp

Table 3-99: Ports

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Output with modified MasterID.
pvbus_s	PVBus	Slave	Unmodified input.

Parameters for LabellerForDMA330_cpp

dma330_data_label

Type

int

Default value

0x0

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

Discriminate between DMA-330 data channels. Channel ID is added to the data label.

dma330_ns_instruction_label

Type
int
Default value
0x0

The label to apply to all non-secure _instructions_ transactions flowing through the labeller.

dma330_s_instruction_label

Type
int
Default value
0x0

The label to apply to all secure _instructions_ transactions flowing through the labeller.

3.3.4 LabellerForGPUProtMode

This model is written in LISA+.

Iris and MTI instances for LabellerForGPUProtMode

This model has the following Iris instances:

Table 3-100: LabellerForGPUProtMode Iris instances

InstanceName	ComponentName
LabellerForGPUProtMode	LabellerForGPUProtMode
LabellerForGPUProtMode.pvbusmodifier	PVBusMapper

This model has the following MTI trace components:

Table 3-101: LabellerForGPUProtMode MTI instances

InstanceName	ComponentName
LabellerForGPUProtMode.pvbusmodifier	PVBusMapper

LabellerForGPUProtMode contains the following CADI targets:

- LabellerForGPUProtMode

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.

Ports for LabellerForGPUProtMode

Table 3-102: Ports

Name	Protocol	Type	Description
<code>prot_mode</code>	Signal	Slave	Input to determine whether output is supposed to be protected or not
<code>pvbuse_m</code>	PVBus	Master	Output with modified MasterID.
<code>pvbuse_s</code>	PVBus	Slave	Unmodified input.

Parameters for LabellerForGPUProtMode

`gpu_id_normal`

Type

int

Default value

0x0

NSAID to apply to all transactions flowing through the labeller when `prot_mode` is low.

`gpu_id_protected`

Type

int

Default value

0x0

NSAID to apply to all transactions flowing through the labeller when `prot_mode` is high.

3.3.5 LabellerForGPUProtMode_cpp

This model is written in C++.

Iris and MTI instances for LabellerForGPUProtMode_cpp

This model has the following Iris instances:

Table 3-103: LabellerForGPUProtMode_cpp Iris instances

InstanceName	ComponentName
LabellerForGPUProtMode_cpp	LabellerForGPUProtMode
LabellerForGPUProtMode_cpp.PVBusModifier	PVBusMapper

This model has the following MTI trace components:

Table 3-104: LabellerForGPUProtMode_cpp MTI instances

InstanceName	ComponentName
LabellerForGPUProtMode_cpp.PVBusModifier	PVBusMapper

LabellerForGPUProtMode_cpp contains the following CADI targets:

- LabellerForGPUProtMode

Ports for LabellerForGPUProtMode_cpp

Table 3-105: 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_cpp

gpu_id_normal

Type

int

Default value

0x0

NSAID to apply to all transactions flowing through the labeller when prot_mode is low.

gpu_id_protected

Type

int

Default value

0x0

NSAID to apply to all transactions flowing through the labeller when `prot_mode` is high.

3.3.6 Labeller_cpp

This model is written in C++.

Iris and MTI instances for Labeller_cpp

This model has the following Iris instances:

Table 3-106: Labeller_cpp Iris instances

InstanceName	ComponentName
Labeller_cpp	Labeller
Labeller_cpp.PVBusModifier	PVBusMapper

This model has the following MTI trace components:

Table 3-107: Labeller_cpp MTI instances

InstanceName	ComponentName
Labeller_cpp.PVBusModifier	PVBusMapper

Labeller_cpp contains the following CADI targets:

- Labeller

Ports for Labeller_cpp

Table 3-108: Ports

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	-
pvbus_s	PVBus	Slave	-

Parameters for Labeller_cpp

label

Type

int

Default value

0x0

The label to apply to all transactions flowing through the labeller.

3.3.7 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+.

Iris and MTI instances for MSIRewriter

This model has the following Iris instances:

Table 3-109: MSIRewriter Iris instances

InstanceName	ComponentName
MSIRewriter	MSIRewriter
MSIRewriter.mapper	PVBusMapper
MSIRewriter.pvbusmaster	PVBusMaster
MSIRewriter.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Table 3-110: MSIRewriter MTI instances

InstanceName	ComponentName
MSIRewriter	MSIRewriter
MSIRewriter.mapper	PVBusMapper
MSIRewriter.pvbusmaster	PVBusMaster
MSIRewriter.pvbusslave	PVBusSlave

MSIRewriter contains the following CADI targets:

- MSIRewriter

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.

Ports for MSIRewriter

Table 3-111: Ports

Name	Protocol	Type	Description
<code>pvbus_m</code>	PVBus	Master	-
<code>pvbus_s</code>	PVBus	Slave	-

Parameters for MSIRewriter

`GITS_TRANSLATE64R_OFFSET`

Type

int

Default value

0x10048

It is an offset from ITS0-Base.

`ITS0-base`

Type

int

Default value

0x0

Register base address for ITS0. This base address is used to recognise writes to the `GITS_TRANSLATER` register within the ITS0's register frame.

`enable_rewriting`

Type

bool

Default value

0x1

Enable rewriting.

`label`

Type

int

Default value

0xffffffff

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

Log level, 0 is off.

3.3.8 PASSwitch

Allow transactions from Realm Management Extension(RME) worlds (realm/root/secure/non_secure) to be routed separately. This model is written in C++.

Ports for PASSwitch

Table 3-112: Ports

Name	Protocol	Type	Description
control	PASSwitchControl	Slave	Controls routing of transactions
pvbus_m[4]	PVBus	Master	Manager ports of PASSwitch
pvbus_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

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

3.3.9 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+.

Iris and MTI instances for PVBUS4KBTo1KBSplitter

This model has the following Iris instances:

Table 3-113: 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 3-114: 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

PVBUS4KBTo1KBSplitter contains the following CADI targets:

- PVBUS4KBTo1KBSplitter

About PVBus4KBTo1KBSplitter

The purpose of this component is to allow an upstream component to access four downstream components in the same 4KB address range. It splits the 4KB range from 0 to 0xfff into the following four 1KB ranges, which allows four different components to be attached to the 4KB 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 4KB in size.



The forwarded transactions have an address re-aligned with 1KB boundary in the range 0, 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.

Ports for PVBus4KBTo1KBSplitter

Table 3-115: Ports

Name	Protocol	Type	Description
pvbus_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.
pvbus_s	PVBus	Slave	The upstream port. Accepts addresses in range 0x0 - 0x0FFF. Outside of this range transactions will abort.

3.3.10 PVBusCache

A PVBusCache manages cache-line data and supports forwarding of transactions. This model is written in C++.

Ports for PVBusCache

Table 3-116: 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/control port.

Name	Protocol	Type	Description
device	PVBusCacheDevice	Master	Connection out to cache device.

3.3.11 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 `pdbus_m_range` port. In LISA+, the syntax for this is:

```
decoder.pdbus_m_range[start..end] = slave.pdbus
```

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> pdbus_s;
    master port<PVBus> sdram;
    master port<PVBus> flash;
    master port<PVBus> uart;

    composition
    {
        pvdecoder : PVBusDecoder;
    }

    connection
    {
        self.pdbus_s => pvdecoder.pdbus_s;
        pvdecoder.pdbus_m_range[0x000000..0x0ffffff] => sdram;
        pvdecoder.pdbus_m_range[0x100000..0x1ffffff] => flash;
        pvdecoder.pdbus_m_range[0x200000..0x2ffffff] => uart;
        pvdecoder.pdbus_m_range[0xff0000..0xffffffff] => sdram[0x070000..0x07ffff];
    }
}
```

Ports for PVBusDecoder

Table 3-117: Ports

Name	Protocol	Type	Description
pdbus_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.
pdbus_s	PVBus	Slave	Accepts incoming transactions. Connect this port to a bus master, or to the output of another bus decoder.

3.3.12 PVBusExclusiveMonitor

Global exclusive monitor. This model is written in C++.

Iris and MTI instances for PVBusExclusiveMonitor

This model has the following Iris instances:

Table 3-118: PVBusExclusiveMonitor Iris instances

InstanceName	ComponentName
PVBusExclusiveMonitor	PVBusExclusiveMonitor
PVBusExclusiveMonitor.bus_mapper	PVBusMapper

This model has the following MTI trace components:

Table 3-119: PVBusExclusiveMonitor MTI instances

InstanceName	ComponentName
PVBusExclusiveMonitor	PVBusExclusiveMonitor
PVBusExclusiveMonitor.bus_mapper	PVBusMapper

PVBusExclusiveMonitor contains the following CADI targets:

- PVBusExclusiveMonitor

Ports for PVBusExclusiveMonitor

Table 3-120: 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

Apply the given exclusive store width matching criteria to non-exclusive stores.

clear_on_strex_address_mismatch

Type

bool

Default value

0x1

Whether monitor is cleared when strex fails due to address mismatch.

enable_component**Type**

bool

Default value

0x1

Enable component.

exclusive_monitor_clear_on_atomic_from_same_master**Type**

bool

Default value

0x1

Monitor atomics from the same master.

log2_granule_size**Type**

int

Default value

0x0

log2 of address granule size.

match_access_width**Type**

bool

Default value

0x0

Fail STREX if not the same access width as LDREX.

match_secure_state**Type**

bool

Default value

0x1

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

Monitor non-exclusive stores from the same master.

number_of_monitors

Type

int

Default value

0x8

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

3.3.13 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 3-121: PVBusExclusiveSquasher Iris instances

InstanceName	ComponentName
PVBusExclusiveSquasher	PVBusExclusiveSquasher
PVBusExclusiveSquasher.bus_modifier	PVBusMapper

This model has the following MTI trace components:

Table 3-122: PVBusExclusiveSquasher MTI instances

InstanceName	ComponentName
PVBusExclusiveSquasher.bus_modifier	PVBusMapper

PVBusExclusiveSquasher contains the following CADI targets:

- PVBusExclusiveSquasher

Ports for PVBusExclusiveSquasher

Table 3-123: Ports

Name	Protocol	Type	Description
pdbus_m	PVBus	Master	Bus master port.
pdbus_s	PVBus	Slave	Bus slave port.

3.3.14 PVBusExclusiveSquasher_cpp

Squashes the exclusive attribute on bus transactions. This model is written in C++.

Iris and MTI instances for PVBusExclusiveSquasher_cpp

This model has the following Iris instances:

Table 3-124: PVBusExclusiveSquasher_cpp Iris instances

InstanceName	ComponentName
PVBusExclusiveSquasher_cpp	PVBusExclusiveSquasher
PVBusExclusiveSquasher_cpp.PVBusModifier	PVBusMapper

This model has the following MTI trace components:

Table 3-125: PVBusExclusiveSquasher_cpp MTI instances

InstanceName	ComponentName
PVBusExclusiveSquasher_cpp.PVBusModifier	PVBusMapper

PVBusExclusiveSquasher_cpp contains the following CADI targets:

- PVBusExclusiveSquasher

Ports for PVBusExclusiveSquasher_cpp

Table 3-126: Ports

Name	Protocol	Type	Description
pdbus_m	PVBus	Master	-
pdbus_s	PVBus	Slave	-

3.3.15 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 3-127: PVBusLogger Iris instances

InstanceName	ComponentName
PVBusLogger	PVBusLogger
PVBusLogger.mapper	PVBusMapper

This model has the following MTI trace components:

Table 3-128: PVBusLogger MTI instances

InstanceName	ComponentName
PVBusLogger	PVBusLogger
PVBusLogger.mapper	PVBusMapper

PVBusLogger contains the following CADI targets:

- PVBusLogger

Ports for PVBusLogger

Table 3-129: Ports

Name	Protocol	Type	Description
pdbus_m	PVBus	Master	Bus master port.
pdbus_s	PVBus	Slave	Bus slave port.

Parameters for PVBusLogger

trace_debug

Type

bool

Default value

0x0

Enable tracing of debug transactions.

trace_snoops

Type

bool

Default value

0x0

Enable tracing of ACE snoop requests.

3.3.16 PVBusMapper

Allow transactions to be remapped arbitrarily. This model is written in C++.

Iris and MTI instances for PVBusMapper

This model has the following Iris instances:

Table 3-130: PVBusMapper Iris instances

InstanceName	ComponentName
PVBusMapper	PVBusMapper

This model has the following MTI trace components:

Table 3-131: PVBusMapper MTI instances

InstanceName	ComponentName
PVBusMapper	PVBusMapper

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/`.

Ports for PVBUSMapper

Table 3-132: Ports

Name	Protocol	Type	Description
control	PVBusMapperControl	Master	Configuration port to determine mappings.
pvbus_m[64]	PVBus	Master	Bus master ports.
pvbus_s	PVBus	Slave	Bus slave port.
reset	Signal	Slave	Reset signal.

3.3.17 PVBusMaster

The `PVBusMaster` subcomponent allows a device to generate PVBus transactions. This model is written in C++.

Iris and MTI instances for PVBusMaster

This model has the following Iris instances:

Table 3-133: PVBusMaster Iris instances

InstanceName	ComponentName
PVBusMaster	PVBusMaster

This model has the following MTI trace components:

Table 3-134: PVBusMaster MTI instances

InstanceName	ComponentName
PVBusMaster	PVBusMaster

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 `pdbus_m` port to its own bus port.

Example:

```
component DmaTransfer
{
  master port<PVBUS> pdbus_m;
  master port<PVTransactionMaster> busmaster_control;
  composition {
    busmaster : PVBUSMaster;
  }
  resources {
    pv::TransactionGenerator* stream_in;
    pv::TransactionGenerator* stream_out;
  }
  connection {
    busmaster.pdbus_m => self.pdbus_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;
    }
  }
}
```

Ports for PVBUSMaster

Table 3-135: Ports

Name	Protocol	Type	Description
control	PVTransactionMaster	Slave	Enables the owning component to instantiate <code>pv::TransactionGenerator</code> objects.
pdbus_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.

3.3.18 PVBusModifier

Allow transactions to be remapped arbitrarily. This model is written in C++.

Iris and MTI instances for PVBusModifier

This model has the following Iris instances:

Table 3-136: PVBusModifier Iris instances

InstanceName	ComponentName
PVBusModifier	PVBusMapper

This model has the following MTI trace components:

Table 3-137: PVBusModifier MTI instances

InstanceName	ComponentName
PVBusModifier	PVBusMapper

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.

Ports for PVBusModifier

Table 3-138: 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.

3.3.19 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 3-139: PVBusRouter Iris instances

InstanceName	ComponentName
PVBusRouter	PVBusRouter
PVBusRouter.mapper	PVBusMapper

This model has the following MTI trace components:

Table 3-140: PVBusRouter MTI instances

InstanceName	ComponentName
PVBusRouter.mapper	PVBusMapper

PVBusRouter contains the following CADI targets:

- PVBusRouter

Ports for PVBusRouter

Table 3-141: Ports

Name	Protocol	Type	Description
control	PVBusRouterControl	Master	Configuration port to determine filters.
pdbus_m[64]	PVBus	Master	Bus master ports.
pdbus_s	PVBus	Slave	Bus slave port.

3.3.20 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++.

Iris and MTI instances for PVBusSlave

This model has the following Iris instances:

Table 3-142: PVBusSlave Iris instances

InstanceName	ComponentName
PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-143: PVBUSSlave MTI instances

InstanceName	ComponentName
PVBUSSlave	PVBUSSlave

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);
    }
}

```

Ports for PVBusSlave

Table 3-144: Ports

Name	Protocol	Type	Description
control	PVBusSlaveControl	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	PVDevice	Master	Passes on requests for peripheral register accesses to permit the owning component to handle the request.
pvbus_s	PVBus	Slave	Handles incoming requests from bus masters.
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.

3.3.21 PVMemoryProtectionEngine

Encrypt memory transactions for each encryption context with an independent key to prevent mismatch access. This model is written in C++.

Iris and MTI instances for PVMemoryProtectionEngine

This model has the following Iris instances:

Table 3-145: PVMemoryProtectionEngine Iris instances

InstanceName	ComponentName
PVMemoryProtectionEngine	PVMemoryProtectionEngine
PVMemoryProtectionEngine.mapper	PVBusMapper

This model has the following MTI trace components:

Table 3-146: PVMemoryProtectionEngine MTI instances

InstanceName	ComponentName
PVMemoryProtectionEngine	PVMemoryProtectionEngine
PVMemoryProtectionEngine.mapper	PVBusMapper

PVMemoryProtectionEngine contains the following CADI targets:

- PVMemoryProtectionEngine

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.

Ports for PVMemoryProtectionEngine

Table 3-147: Ports

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Manager ports of the MPE
pvbus_s	PVBus	Slave	Subordinate port of the MPE

Parameters for PVMemoryProtectionEngine

block_size_in_bytes

Type

int

Default value

0x1000

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

Enables the Memory Protection Engine.

ignore_mecid

Type

bool

Default value

0x0

Ignore MECID during encryption key calculation.

non_secure_pas_enc_key

Type

int

Default value

0x22

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"

Encoding of various attributes on the bus.

realm_pas_enc_key

Type

int

Default value

0x88

Realm PAS encryption key.

root_pas_enc_key**Type**

int

Default value

0x44

Root PAS encryption key.

secure_pas_enc_key**Type**

int

Default value

0x11

Secure PAS encryption key.

3.3.22 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 3-148: PVWriteBuffer Iris instances

InstanceName	ComponentName
PVWriteBuffer	PVWriteBuffer
PVWriteBuffer.mapper	PVBusMapper

This model has the following MTI trace components:

Table 3-149: PVWriteBuffer MTI instances

InstanceName	ComponentName
PVWriteBuffer	PVWriteBuffer
PVWriteBuffer.mapper	PVBusMapper

Ports for PVWriteBuffer**Table 3-150: Ports**

Name	Protocol	Type	Description
barrier_notify_s	PVWriteBuffer_BarrierPort	Slave	Barrier notification input.
clk_in	ClockSignal	Slave	Clock input.
pvbuss_m	PVBus	Master	Master connection to memory bus.
pvbuss_s	PVBus	Slave	Slave connection for transactions to be buffered.

Name	Protocol	Type	Description
reset_in	Signal	Slave	Reset input.
serror_notify_m	PVWriteBuffer_SErrorPort	Master	SError output generation.

3.3.23 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 3-151: 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 3-152: SimplePVBUSMaster MTI instances

InstanceName	ComponentName
SimplePVBUSMaster.pvbusmaster	PVBusMaster
SimplePVBUSMaster.pvbusslave	PVBusSlave

SimplePVBUSMaster contains the following CADI targets:

- ClockTimerThread64
- SchedulerThread
- SchedulerThreadEvent
- SimplePVBUSMaster

Ports for SimplePVBUSMaster

Table 3-153: Ports

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Output of generated transactions.
pvbus_s	PVBus	Slave	-

Parameters for SimplePVBusMaster

verbose

Type

bool

Default value

0x0

verbose.

3.3.24 TZSwitch

Allow TrustZone secure/normal bus signals to be routed separately. This model is written in LISA+.

Iris and MTI instances for TZSwitch

This model has the following Iris instances:

Table 3-154: TZSwitch Iris instances

InstanceName	ComponentName
TZSwitch	TZSwitch
TZSwitch.pvbus_mapper	PVBusMapper

This model has the following MTI trace components:

Table 3-155: TZSwitch MTI instances

InstanceName	ComponentName
TZSwitch.pvbus_mapper	PVBusMapper

TZSwitch contains the following CADI targets:

- TZSwitch

About TZSwitch

Allow TrustZone secure/normal bus signals to be routed separately.

Transactions received on the pvbus_input slave port will be 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.

The control port behaviour routeAccesses () takes two arguments.

input selects which types of signals will be 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 will be routed:

TZROUTE_IGNORE

transactions will be 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 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 will be overridden by any runtime calls to `routeAccesses()` on the control port.

Ports for TZSwitch

Table 3-156: Ports

Name	Protocol	Type	Description
<code>control</code>	<code>TZSwitchControl</code>	Slave	Controls routing of transactions.
<code>pvbus_input</code>	<code>PVBus</code>	Slave	Slave port for connection to PVBus master/decoder.
<code>pvbus_port_a</code>	<code>PVBus</code>	Master	Output port a.
<code>pvbus_port_b</code>	<code>PVBus</code>	Master	Output port b.

Parameters for TZSwitch

normal**Type**

int

Default value

0x2

Normal Port.

secure**Type**

int

Default value

0x1

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++.

Iris and MTI instances for ClockDivider

This model has the following Iris instances:

Table 3-157: ClockDivider Iris instances

InstanceName	ComponentName
ClockDivider	ClockDivider

This model has the following MTI trace components:

Table 3-158: ClockDivider MTI instances

InstanceName	ComponentName
ClockDivider	ClockDivider

ClockDivider contains the following CADI targets:

- ClockDivider

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.

Ports for ClockDivider

Table 3-159: Ports

Name	Protocol	Type	Description
<code>clk_in</code>	ClockSignal	Slave	Input clock signal, coming from a MasterClock or another ClockDivider.
<code>clk_out</code>	ClockSignal	Master	Clock signal generated by this ClockDivider.
<code>rate</code>	ClockRateControl	Slave	Permits you to dynamically change the clock divider ratio.

Parameters for ClockDivider

div

Type

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

mul

Type

int

Default value

0x1

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 3-160: ClockGate Iris instances

InstanceName	ComponentName
ClockGate	ClockGate
ClockGate.ClockDivider	ClockDivider

This model has the following MTI trace components:

Table 3-161: ClockGate MTI instances

InstanceName	ComponentName
ClockGate.ClockDivider	ClockDivider

ClockGate contains the following CADI targets:

- ClockGate

Ports for ClockGate

Table 3-162: 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

Diagnostics.

divider.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

divider.mul**Type**

int

Default value

0x1

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 3-163: ClockSelector Iris instances

InstanceName	ComponentName
ClockSelector	ClockSelector
ClockSelector.ClockDivider	ClockDivider

This model has the following MTI trace components:

Table 3-164: ClockSelector MTI instances

InstanceName	ComponentName
ClockSelector.ClockDivider	ClockDivider

ClockSelector contains the following CADI targets:

- ClockSelector

Ports for ClockSelector

Table 3-165: Ports

Name	Protocol	Type	Description
clk_in[11]	ClockSignal	Slave	-
clk_out	ClockSignal	Master	-

Name	Protocol	Type	Description
clk_sel	Signal	Slave	-
clk_sel_num	Value	Slave	-

Parameters for ClockSelector

clkdiv0.div

Type

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv0.mul

Type

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv1.div

Type

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv1.mul

Type

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv10.div

Type

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv10.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv2.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv2.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv3.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv3.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv4.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv4.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv5.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv5.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv6.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv6.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv7.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv7.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv8.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv8.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv9.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv9.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkdivider.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkdivider.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

diagnostics**Type**

int

Default value

0x0

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 3-166: Ports

Name	Protocol	Type	Description
<code>clk_in</code>	<code>ClockSignal</code>	Slave	Determines the tick rate of the timer.
<code>timer_callback</code>	<code>TimerCallback</code>	Master	Port on which a signal is sent after the number of scheduled ticks has elapsed.
<code>timer_control</code>	<code>TimerControl</code>	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 3-167: Ports

Name	Protocol	Type	Description
<code>clk_in</code>	<code>ClockSignal</code>	Slave	Determines the tick rate of the timer.
<code>timer_callback</code>	<code>TimerCallback64</code>	Master	Port on which a signal is sent after the number of scheduled ticks has elapsed.
<code>timer_control</code>	<code>TimerControl64</code>	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+.

Iris and MTI instances for `ClockTimerThread`

This model has the following Iris instances:

Table 3-168: `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>

`ClockTimerThread` contains the following CADI targets:

- `ClockTimerThread`
- `ClockTimerThread64`
- `SchedulerThread`
- `SchedulerThreadEvent`

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)`.

Ports for ClockTimerThread

Table 3-169: Ports

Name	Protocol	Type	Description
<code>clk_in</code>	<code>ClockSignal</code>	Slave	Determines the tick rate of the timer.
<code>timer_callback</code>	<code>TimerCallback</code>	Master	Port on which a signal is sent after the number of scheduled ticks has elapsed.
<code>timer_control</code>	<code>TimerControl</code>	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+.

Iris and MTI instances for ClockTimerThread64

This model has the following Iris instances:

Table 3-170: ClockTimerThread64 Iris instances

InstanceName	ComponentName
<code>ClockTimerThread64</code>	<code>ClockTimerThread64</code>
<code>ClockTimerThread64.thread</code>	<code>SchedulerThread</code>
<code>ClockTimerThread64.thread_event</code>	<code>SchedulerThreadEvent</code>

`ClockTimerThread64` contains the following CADI targets:

- `ClockTimerThread64`
- `SchedulerThread`
- `SchedulerThreadEvent`

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)`.

Ports for ClockTimerThread64

Table 3-171: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Determines the tick rate of the timer.
timer_callback	TimerCallback64	Master	Port on which a signal is sent after the number of scheduled ticks has elapsed.
timer_control	TimerControl64	Slave	Permits the timer to be set, canceled and queried.

3.4.8 ClockTimerThread64_cpp

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the sg::Signal() callback from a proper scheduler thread. This mean that the sg::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 which issue bus transactions from within the timer sg::Signal() callback must use ClockTimerThread(64) rather than ClockTimer(64). This model is written in C++.

Iris and MTI instances for ClockTimerThread64_cpp

This model has the following Iris instances:

Table 3-172: ClockTimerThread64_cpp Iris instances

InstanceName	ComponentName
ClockTimerThread64_cpp	ClockTimerThread64
ClockTimerThread64_cpp.SchedulerThread	SchedulerThread
ClockTimerThread64_cpp.SchedulerThreadEvent	SchedulerThreadEvent

ClockTimerThread64_cpp contains the following CADI targets:

- ClockTimerThread64

Ports for ClockTimerThread64_cpp

Table 3-173: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-
timer_callback	TimerCallback64	Master	-
timer_control	TimerControl64	Slave	-

3.4.9 ClockTimerThread_cpp

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 sg::Signal() callback from a proper scheduler thread. This mean that the sg::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 which issue bus transactions from within the timer

sg::Signal() callback must use ClockTimerThread(64) rather than ClockTimer(64). This model is written in C++.

Iris and MTI instances for ClockTimerThread_cpp

This model has the following Iris instances:

Table 3-174: ClockTimerThread_cpp Iris instances

InstanceName	ComponentName
ClockTimerThread_cpp	ClockTimerThread
ClockTimerThread_cpp.ClockTimerThread64	ClockTimerThread64
ClockTimerThread_cpp.ClockTimerThread64.SchedulerThread	SchedulerThread
ClockTimerThread_cpp.ClockTimerThread64.SchedulerThreadEvent	SchedulerThreadEvent

ClockTimerThread_cpp contains the following CADI targets:

- ClockTimerThread

Ports for ClockTimerThread_cpp

Table 3-175: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-
timer_callback	TimerCallback	Master	-
timer_control	TimerControl	Slave	-

3.4.10 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.

A system might contain more than one MasterClock, all of which generate the same clockSignal rate.

Ports for MasterClock

Table 3-176: Ports

Name	Protocol	Type	Description
clk_out	ClockSignal	Master	Master clock rate.

3.4.11 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 3-177: PLLControl Iris instances

InstanceName	ComponentName
PLLControl	PLLControl
PLLControl.ClockDivider	ClockDivider

This model has the following MTI trace components:

Table 3-178: PLLControl MTI instances

InstanceName	ComponentName
PLLControl.ClockDivider	ClockDivider

PLLControl contains the following CADI targets:

- PLLControl

Ports for PLLControl

Table 3-179: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-
clk_out	ClockSignal	Master	-
lock	Signal	Master	-
rate	ClockRateControl	Slave	-
unlock	Signal	Master	-

Parameters for PLLControl

clkdiv.div

Type

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkdiv.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

3.4.12 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 3-180: ScalableClockControl Iris instances

InstanceName	ComponentName
ScalableClockControl	ScalableClockControl
ScalableClockControl.ClockDivider	ClockDivider
ScalableClockControl.ClockGate	ClockGate
ScalableClockControl.ClockGate.ClockDivider	ClockDivider
ScalableClockControl.ClockSelector	ClockSelector
ScalableClockControl.ClockSelector.ClockDivider	ClockDivider

This model has the following MTI trace components:

Table 3-181: ScalableClockControl MTI instances

InstanceName	ComponentName
ScalableClockControl.ClockDivider	ClockDivider
ScalableClockControl.ClockGate.ClockDivider	ClockDivider
ScalableClockControl.ClockSelector.ClockDivider	ClockDivider

ScalableClockControl contains the following CADI targets:

- ScalableClockControl

Ports for ScalableClockControl

Table 3-182: 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

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv1.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv10.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv10.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv2.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv2.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv3.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv3.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv4.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv4.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv5.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv5.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv6.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv6.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv7.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv7.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv8.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv8.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv9.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv9.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkGate.diagnostics**Type**

int

Default value

0x0

Diagnostics.

clkGate.divider.div**Type**

int

Default value

0x1

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkSelect.diagnostics

Type
int
Default value
0x0

Diagnostics.

3.4.13 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 3-183: SwitchedClockControl Iris instances

InstanceName	ComponentName
SwitchedClockControl	SwitchedClockControl

InstanceName	ComponentName
SwitchedClockControl.ClockDivider	ClockDivider
SwitchedClockControl.ClockGate	ClockGate
SwitchedClockControl.ClockGate.ClockDivider	ClockDivider
SwitchedClockControl.ClockSelector	ClockSelector
SwitchedClockControl.ClockSelector.ClockDivider	ClockDivider

This model has the following MTI trace components:

Table 3-184: SwitchedClockControl MTI instances

InstanceName	ComponentName
SwitchedClockControl.ClockDivider	ClockDivider
SwitchedClockControl.ClockGate.ClockDivider	ClockDivider
SwitchedClockControl.ClockSelector.ClockDivider	ClockDivider

SwitchedClockControl contains the following CADI targets:

- SwitchedClockControl

Ports for SwitchedClockControl

Table 3-185: 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

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv1.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv2.div**Type**

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

clkDiv2.mul**Type**

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkGate.diagnostics**Type**

int

Default value

0x0

Diagnostics.

clkGate.divider.div**Type**

int

Default value

0x1

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clkSelect.diagnostics**Type**

int

Default value

0x0

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 3-186: 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 23.

Iris and MTI instances for AEMv8RMPCT

This model has the following Iris instances:

Table 3-187: 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

InstanceName	ComponentName
AEMv8RMPCT.cpu0.dtlb	TlbCadi
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
AEMv8RMPCT.l2_cache.upstream[0]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[10]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[11]	PVBusSlave
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

This model has the following MTI trace components:

Table 3-188: AEMv8RMPCT MTI instances

InstanceName	ComponentName
AEMv8RMPCT	ARMv8Cluster
AEMv8RMPCT.AMU	PVBusLogger
AEMv8RMPCT.AMU.mapper	PVBusMapper
AEMv8RMPCT.DAP	PVBusLogger
AEMv8RMPCT.DAP.mapper	PVBusMapper
AEMv8RMPCT.MMAP	PVBusLogger
AEMv8RMPCT.MMAP.mapper	PVBusMapper
AEMv8RMPCT.RAS	PVBusLogger

InstanceName	ComponentName
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
AEMv8RMPCT.l2_cache.upstream[0]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[10]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[11]	PVBusSlave
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 contains the following CADI targets:

- ARMAEMv8-R_MP
- Cluster_ARMAEMv8-R_MP
- PVCache
- TlbCadi

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 3-189: 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
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](#).

Ports for AEMv8RMPCT

Table 3-190: 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.

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
CNTHPSIRQ[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 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	-
ctilextin[4]	Signal	Slave	-
ctilextout[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.

Name	Protocol	Type	Description
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.
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).

Name	Protocol	Type	Description
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for AEMv8RMPCT

ADFSR-AIFSR-implemented

Type

bool

Default value

0x0

ADFSR and AIFSR are implemented.

AIDR

Type

int

Default value

0x0

Value of AIDR_EL1 register.

AMIIDR

Type

int

Default value

0x43b

Value of AMU Implementation Identification Register.

AMPIDR

Type

int

Default value

0x4000bb000

Value of AMU Peripheral Identification Register.

BPIMVA_causes_translation_lookup

Type

bool

Default value

0x0

Do a translation when BPIMVA instruction is executed (which may cause a translation fault).

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation to NC/DEV memory. The broadcastatomic signal will override this value if used.

BROADCASTATOMICL

Type

bool

Default value

0x1

Enable broadcasting of atomic operation to LLRAM memory. The broadcastatomicl signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTINNER

Type

bool

Default value

0x1

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CCSIDR-L1D_override**Type**

int

Default value

0x0

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

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

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

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

CFGTFPEN Configuration pin at reset for bitfield IMP_MEMPROTCTLR_EL1.TFPEN.

CHI**Type**

bool

Default value

0x0

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Value of CPU Configuration Register.

CTIPIDR**Type**

int

Default value

0x0

If non-zero, override the CTI Peripheral Identification Register.

CTR-L1Ip-override**Type**

int

Default value

0x0

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

If true, RES0 behaviour is applied to DBGBCR(_EL1).BT before checking for reserved values for this field.

DBGPIDR**Type**

int

Default value

0x0

If non-zero, override the Debug Peripheral Identification Register.

DBGROMADDR**Type**

int

Default value

0x0

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x0

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

ERRIIDR**Type**

int

Default value

0xd800143b

Value of RAS Implementation Identification Register.

ERRPIDR**Type**

int

Default value

0x4100bbd80

Value of RAS Peripheral Identification Register.

ERXMISC0_mask**Type**

int

Default value

0x0

Write Mask for ERXMISC0 RAS Register.

GICDISABLE**Type**

bool

Default value

0x1

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

Value of MIDR_EL1 register.

NUM_CORES**Type**

int

Default value

0x1

Number of cores in cluster.

PA_SIZE**Type**

int

Default value

0x28

Physical address range supported (FEAT_LPA).

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

PMCEID0**Type**

int

Default value

0xffffffff

Performance Monitor Common Event ID Reg 0 value - 64 bit.

PMCEID1**Type**

int

Default value

0xffffffff

Performance Monitor Common Event ID Reg 1 value - 64 bit.

PMSIDR.ArchInst**Type**

bool

Default value

0x1

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

Defines whether call return branch records (FEAT_SPE_CRR) is implemented or not.

PMSIDR.LDS**Type**

bool

Default value

0x0

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

If non-zero, override the PMU Peripheral Identification Register.

VMSA_supported**Type**

bool

Default value

0x1

VMSA is supported at EL1.

abort_execution_from_device_memory**Type**

bool

Default value

0x0

Execution from device memory generates a prefetch abort.

advsimd_overread**Type**

bool

Default value

0x0

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

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

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

Align PC when performing an illegal exception return from AArch64 to AArch32.

amu_aux_type_fixed**Type**

string

Default value

""

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

""

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

Number of AMU auxiliary counters implemented.

apsr_read_restrict**Type**

bool

Default value

0x0

At EL0, unknown bits of APSR are RAZ.

atomic_memtype_fault_priority

Type

int

Default value

0x0

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

Value of AFR0 ID register.

branch-predictor-clear-policy

Type

int

Default value

0x2

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

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

Enable TCM, L1Cache bus protection after reset.

cache-log2linelen**Type**

int

Default value

0x6

Log2 of the cache line length in bytes.

cache_maintenance_hits_watchpoints**Type**

bool

Default value

0x0

DCIMVA operations executed in AArch32 modes hit watchpoints.

changing_block_size_without_bbm_support**Type**

int

Default value

0x0

Level of support for changing block size without break-before-make (FEAT_BBM).

check_memory_attributes**Type**

bool

Default value

0x1

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

Clean and invalidate caches on warm reset.

clear_reg_top_eret**Type**

int

Default value

0x1

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

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

Unique cluster transaction identifier for interconnect protection.

configure_pmu_events_with_json**Type**

string

Default value

""

"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

""

"[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

""

"[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

""

"[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

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

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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.CP15SDISABLE2**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers (FEAT_CP15SDISABLE2).

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.DCZID-log2-block-size**Type**

int

Default value

0x8

Log2 of the block size cleared by DC ZVA instruction (as read from DCZID_ELO).

cpuX.DCZVA_single_write**Type**

bool

Default value

0x0

Execute the DCZVA as a single write.

cpuX.MPIDR-override**Type**

int

Default value

0x0

Override of MPIDR value. If nonzero will override the MT, cluster and CPU ID bits in MPIDR.

cpuX.RVBAR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.RVBAR32**Type**

int

Default value

0x0

Reset vector address in AARCH32 when VINITHI is not set and ignore_rvbar_in_aarch32 is set.

cpuX.SMPnAMP**Type**

bool

Default value

0x1

Enable broadcast messages necessary for correct SMP operation at reset.

cpuX.TEINIT**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTL.R.V.

cpuX.aarch32_reset_from_impdef_addr**Type**

bool

Default value

0x1

If PE resets into AArch32, Whether execution starts from IMPDEF address or hi/low vector.

cpuX.ase-present**Type**

bool

Default value

0x1

Set whether the model has been built with NEON support.

cpuX.clock_divider**Type**

int

Default value

0x1

Clock divider ratio for asymmetric MP clocking.

cpuX.clock_multiplier**Type**

int

Default value

0x1

Clock divider ratio for asymmetric MP clocking.

cpuX.crypto_aes**Type**

int

Default value

0x2

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

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

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

Set bits represent that the corresponding trigger requires software acknowledge via CTIINTACK.

cpuX.cti-number_of_claim_bits**Type**

int

Default value

0x0

Number of implemented bits in CTICLAIMSET.

cpuX.cti-number_of_triggers**Type**

int

Default value

0x8

Number of cti event triggers (default: 8, valid values: {3, 8-32}).

cpuX.dcache-size**Type**

int

Default value

0x8000

L1 D-Cache size in bytes.

cpuX.enable_crc32**Type**

int

Default value

0x0

CRC32 instructions supported. 0, not implemented. 1, CRC32 instructions implemented (FEAT_CRC32).

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.etm-present**Type**

bool

Default value

0x1

Set whether the model has ETM support.

cpuX.flash.enable**Type**

bool

Default value

0x0

Enable flash by default after reset.

cpuX.force-fpsid**Type**

bool

Default value

0x0

Override the FPSID value.

cpuX.force-fpsid-value**Type**

int

Default value

0x0

Value to override the FPSID value to.

cpuX.has_hcptr_tase**Type**

bool

Default value

0x1

If false, HCPTR.TASE is RES0.

cpuX.icache-size**Type**

int

Default value

0x8000

L1 I-Cache size in bytes.

cpuX.llpp.base**Type**

int

Default value

0x0

Sets the base address of Low Latency Peripheral Port.

cpuX.llpp.size**Type**

int

Default value

0x1000

Sets the size of LLPP(in bytes).

cpuX.max_code_cache_mb**Type**

int

Default value

0x100

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.number-of-breakpoints**Type**

int

Default value

0x10

Number of breakpoints.

cpuX.number-of-context-breakpoints**Type**

int

Default value

0x10

Number of breakpoints that are context aware.

cpuX.number-of-watchpoints**Type**

int

Default value

0x10

Number of watchpoints.

cpuX.operation_bandwidth**Type**

int

Default value

0x1

Operation width for ARMv8.4 PMU extension.

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-prefix**Type**

bool

Default value

0x0

Prefix semihosting output with target instance name.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.semihosting-stderr_istty**Type**

bool

Default value

0x1

Result for semihost istry call when argument is stderr.

cpuX.semihosting-stdin_istty**Type**

bool

Default value

0x1

Result for semihost istry call when argument is stdin.

cpuX.semihosting-stdout_istty**Type**

bool

Default value

0x1

Result for semihost istry call when argument is stdout.

cpuX.semihosting-use_stderr**Type**

bool

Default value

0x0

Send stderr from the simulated process to host stderr.

cpuX.tcm-present**Type**

bool

Default value

0x0

Disables the TCMs.

cpuX.tcm-supports-exclusive**Type**

bool

Default value

0x0

Whether TCM supports exclusive access.

cpuX.tcm.a.base**Type**

int

Default value

0x0

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

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

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

Whether ATCM clock stretched to occupy full cycle.

cpuX.tcm.a.wait**Type**

int

Default value

0x0

ATCM accesses wait states.

cpuX.tcm.b.base**Type**

int

Default value

0x0

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

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

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

Whether BTCM clock stretched to occupy full cycle.

cpuX.tcm.b.wait**Type**

int

Default value

0x0

BTCLM accesses wait states.

cpuX.tcm.c.base**Type**

int

Default value

0x0

Sets the base address of the CTCLM. CTCLMs are implementation defined in the v8R architecture. CTCLMs are configurable on the model via parameter only.

cpuX.tcm.c.enable**Type**

bool

Default value

0x0

Enable CTCLM by default after reset. CTCLMs are implementation defined in the v8R architecture. CTCLMs are configurable on the model via parameter only.

cpuX.tcm.c.size**Type**

int

Default value

0x2000

Sets the size of the CTCLM(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

Whether CTCLM clock stretched to occupy full cycle.

cpuX.tcm.c.wait**Type**

int

Default value

0x0

CTCM accesses wait states.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Whether double-precision floating point feature is implemented (FEAT_F64MM).

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

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

Set whether the model has VFP support.

cpuX.vfp-traps**Type**

bool

Default value

0x1

Implement support for trapping floating-point exceptions.

cpuX.vfp-traps-show-all**Type**

bool

Default value

0x0

Report all trapped floating-point exceptions in the syndrome when a combination occurs.

cpuX.wfet_early_or_delayed_timeout**Type**

int

Default value

0x0

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

WFIT early or delayed timeout beyond the threshold value of CNTVCT_ELO in percentage.

dbg-bcr-reserved-behavior**Type**

int

Default value

0x1

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

Debug ROM dap base address.

dbgitr_buffer_size**Type**

int

Default value

0x0

Number of instructions which can be buffered before EDSCR.ITE is cleared.

dbgxvr_ress_is_stateful**Type**

bool

Default value

0x0

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

L1 D-Cache read bus width in bytes used to calculate per-access timing annotations.

`dcache-read_latency`

Type

int

Default value

0x0

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

L1 D-Cache size in bytes.

`dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-ways**Type**

int

Default value

0x2

L1 D-Cache number of ways (sets are implicit from size).

dcache-write_access_latency**Type**

int

Default value

0x0

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

L1 D-Cache write bus width in bytes used to calculate per-access timing annotations.

dcache-write_latency**Type**

int

Default value

0x0

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

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

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

""

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

""

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

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

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

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" }

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

""

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

shareability for default memory map regions which are shareable.

delay_serror**Type**

int

Default value

0x0

Add a propagation delay of serror signal into the core.

dic-spi_count**Type**

int

Default value

0x40

Number of shared peripheral interrupts implemented.

disable_impdef_abort_on_ic_maintenance**Type**

bool

Default value

0x0

Disable IMP_INTLATENCY_EL2.MMDVM/LLRAMDVM controlling the abort on IC maintenance on MM port/LLRAM.

disable_sve_plugin**Type**

bool

Default value

0x0

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

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

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

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

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

Undef when secure debug is disabled (EDSCR.SDD == 1) && boolean
IMPLEMENTATION_DEFINED 'EL3 trap priority when SDD == 1'.

enable_address_contig_check**Type**

bool

Default value

0x0

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

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

Whether the core is configured in Dual Core Lock Step mode (FEAT_DCLS).

enable_tlb_contig_check**Type**

bool

Default value

0x0

Perform extra pagetable walks to check translation table entries that have the contiguous hint bit set.

enhanced_pac2_level**Type**

int

Default value

0x0

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

""

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

""

File path to the RAS feature register values as JSON. The file uses the same format as the error_record_feature_register parameter value.

erxpgctl_res0_stateful_mask**Type**

int

Default value

0x0

Mask for stateful bits for ERXPGCTL which are RES0.

exception_catch_before_software_step**Type**

bool

Default value

0x1

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

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

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

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

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

Exclusive monitors in the cluster will be cleared when a strex succeeds.

exercise_stxr_fail**Type**

bool

Default value

0x0

Reject a pseudo-random majority of exclusive store instructions.

ext_abort_device_GRE_prefetch_ras_index**Type**

int

Default value

0xffffffffffffffff

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

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

Critical reporting of device-GRE read external aborts.

ext_abort_device_GRE_read_is_sync**Type**

int

Default value

0x2

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

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

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

Critical reporting of device-GRE write external aborts.

ext_abort_device_GRE_write_is_sync**Type**

int

Default value

0x2

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

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

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

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

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

Critical reporting of device-nGRE read external aborts.

ext_abort_device_nGRE_read_is_sync**Type**

int

Default value

0x2

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

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

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

Critical reporting of device-nGRE write external aborts.

ext_abort_device_nGRE_write_is_sync**Type**

int

Default value

0x2

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

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

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

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

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

Synchronous reporting of device read with acquire external aborts.

ext_abort_device_read_is_critical**Type**

bool

Default value

0x0

Critical reporting of device-nGnRE read external aborts.

ext_abort_device_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_read_ras_index**Type**

int

Default value

0x0

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

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

Critical reporting of device-nGnRE write external aborts.

ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_device_write_ras_index**Type**

int

Default value

0x0

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

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

0xfdfdfdfcfdfdfdf

Returned data, if external aborts are asynchronous.

ext_abort_normal_cacheable_read_is_critical**Type**

bool

Default value

0x0

Critical reporting of normal write-back cacheable-read external aborts.

ext_abort_normal_cacheable_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of normal write-back cacheable-read external aborts.

ext_abort_normal_cacheable_read_ras_index**Type**

int

Default value

0x0

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

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

Critical reporting of normal write-back cacheable write external aborts.

ext_abort_normal_cacheable_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of normal write-back cacheable write external aborts.

ext_abort_normal_cacheable_write_ras_index**Type**

int

Default value

0x0

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

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

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

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

Critical reporting of normal noncacheable-read external aborts.

ext_abort_normal_noncacheable_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of normal noncacheable-read external aborts.

ext_abort_normal_noncacheable_read_ras_index**Type**

int

Default value

0x0

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

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

Critical reporting of normal noncacheable write external aborts.

ext_abort_normal_noncacheable_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of normal noncacheable write external aborts.

ext_abort_normal_noncacheable_write_ras_index**Type**

int

Default value

0x0

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

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

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

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

Critical reporting of normal write-through cacheable-read external aborts.

ext_abort_normal_wt_cacheable_read_is_sync**Type**

int

Default value

0x2

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

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

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

Critical reporting of normal write-through write external aborts.

ext_abort_normal_wt_cacheable_write_is_sync**Type**

int

Default value

0x2

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

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

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

Critical reporting of external aborts generated by device-GRE instruction fetches.

ext_abort_prefetch_device_GRE_read_is_sync**Type**

int

Default value

0x2

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

Critical reporting of external aborts generated by device-nGRE instruction fetches.

ext_abort_prefetch_device_nGRE_read_is_sync**Type**

int

Default value

0x2

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

Critical reporting of external aborts generated by device-nGnRE instruction fetches.

ext_abort_prefetch_device_read_is_sync**Type**

int

Default value

0x2

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

Critical reporting of external aborts generated by normal writeback cacheable instruction fetches.

ext_abort_prefetch_is_sync**Type**

bool

Default value

0x1

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

Critical reporting of external aborts generated by normal noncacheable instruction fetches.

ext_abort_prefetch_noncacheable_read_is_sync**Type**

int

Default value

0x2

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

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

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

Critical reporting of external aborts generated by device-nGnRnE instruction fetches.

ext_abort_prefetch_so_read_is_sync

Type

int

Default value

0x2

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

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

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

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

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

Critical reporting of device-nGnRnE read external aborts.

ext_abort_so_read_is_sync

Type

bool

Default value

0x1

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_read_ras_index

Type

int

Default value

0x0

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

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

Critical reporting of device-nGnRnE write external aborts.

ext_abort_so_write_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of device-nGnRnE write external aborts.

ext_abort_so_write_ras_index**Type**

int

Default value

0x0

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

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

Critical reporting of TTW cacheable read external aborts.

ext_abort_ttw_cacheable_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of TTW cacheable read external aborts.

ext_abort_ttw_cacheable_read_ras_index**Type**

int

Default value

0x0

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

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

Critical reporting of TTW noncacheable read external aborts.

ext_abort_ttw_noncacheable_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of TTW noncacheable read external aborts.

ext_abort_ttw_noncacheable_read_ras_index**Type**

int

Default value

0x0

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

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

Critical reporting of TTW write-through cacheable read external aborts.

ext_abort_ttw_wt_cacheable_read_is_sync**Type**

int

Default value

0x2

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

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

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

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

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

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

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

Enable flash memory protection after reset.

force_align_pc**Type**

bool

Default value

0x0

UNPREDICTABLE branch to non-word-aligned address in ARM state is forced to be aligned.

fpcr_short_vector_raz**Type**

bool

Default value

0x0

FPSCR and FPCR fields LEN and STRIDE are hardwired to 0.

fpsr_res0_stateful_mask**Type**

int

Default value

0x0

Mask for stateful bits of FPSR which are RES0.

gic.GICC-offset**Type**

int

Default value

0x2000

Offset from PERIPHBASE for GICC registers.

gic.GICD-offset**Type**

int

Default value

0x1000

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

Offset from PERIPHBASE for GICH registers.

gic.GICH-other-CPU-offset**Type**

int

Default value

0x5000

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

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

Offset from PERIPHBASE for GICV registers.

gic.PERIPH-size**Type**

int

Default value

0x8000

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

GICv2 compatibility is not supported and GICD_CTLR.ARE_* is always one.

gic_iri.DPG-ARE-only**Type**

bool

Default value

0x0

Limit application of DPG bits to interrupt groups for which ARE=1.

gic_iri.DPG-bits-implemented**Type**

bool

Default value

0x0

Enable implementation of interrupt group participation bits or DPG bits in GICR_CTLR.

gic_iri.GICD-alias**Type**

int

Default value

0x0

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

If true, the GICD_ITARGETS registers are RAZ/WI.

gic_iri.GICD_PIDR**Type**

int

Default value

0x0

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

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

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

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

GICD_IIDR and GICR_IIDR value.

`gic_iri.IRI-ID-bits`**Type**

int

Default value

0x10

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

Number of Interrupt Translation Services to be instantiated (0=none).

`gic_iri.SPI-count`**Type**

int

Default value

0x20

Number of SPIs that are implemented.

`gic_iri.SPI-message-based-support`**Type**

bool

Default value

0x1

Distributor supports message based signaling of SPI.

`gic_iri.STATUSR-implemented`**Type**

bool

Default value

0x0

Determines whether the GICR_STATUSR register is implemented.

`gic_iri.enable_protocol_checking`**Type**

bool

Default value

0x0

Enable/disable protocol checking at cpu interface.

`gic_iri.enabled`**Type**

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

`gic_iri.has-two-security-states`**Type**

bool

Default value

0x0

If true, has two security states.

`gic_iri.irouter-default-mask`**Type**

string

Default value

0.0.0.7

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

Default Reset Value of IROUTER[32..1019] register in the form 'a.b.c.d' or '*'.

`gic_iri.monolithic`**Type**

bool

Default value

0x1

Indicate that the implementation is not distributed.

`gic_iri.non-ARE-core-count`**Type**

int

Default value

0x4

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

Size in bytes allocated to internal GIC Distributor.

`gic_iri.priority-bits`**Type**

int

Default value

0x5

Number of implemented priority bits.

`gic_iri.processor-numbers`**Type**

string

Default value

""

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

Offset from reg-offset where the Redistributors are accessible.

`gic_iri.redistributor-size`**Type**

int

Default value

0x0

Per Redistributor register space in bytes.

`gic_iri.reg-offset`**Type**

int

Default value

0x0

Offset from PERIPHBASE allocated to internal GIC Distributor.

`gic_iri.supports-shareability`**Type**

bool

Default value

0x0

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

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.

`gic_iri.wakeup-on-reset`**Type**

bool

Default value

0x0

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

Whether a non-zero value for affinity at level 3 is supported.

`gicv3.BPR-min`**Type**

int

Default value

0x2

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

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

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

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

Ignore EOI writes that cannot end the highest priority active interrupt.

gicv3.FIQEn-RAO**Type**

bool

Default value

0x0

GICC_CTLR.FIQEn is read as one, write insensitive.

gicv3.IIDR_base**Type**

int

Default value

0x43b

The base value for calculating the GICC_IIDR register value.

gicv3.LR-count**Type**

int

Default value

0x10

The number of implemented list registers.

gicv3.PMHE-RAO-WI**Type**

bool

Default value

0x0

ICC_CTLR_EL*.PHME is read as one, write insensitive.

gicv3.PMHE-RAZ-WI**Type**

bool

Default value

0x0

ICC_CTLR_EL*.PHME is read as zero, write insensitive.

gicv3.PMHE-release-set-packet**Type**

bool

Default value

0x0

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

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

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

Restrict SRE EL3 to be set only once.

gicv3.SRE-enable-action-on-mmap**Type**

int

Default value

0x0

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

If GICv3 CPU interface is being used, this determines whether the STATUS registers are implemented.

gicv3.VBPR-min**Type**

int

Default value

0x2

The minimum value for the GICV_BPR register (non-secure version will be 1 + this value).

gicv3.VFIQEn-RAO**Type**

bool

Default value

0x0

ICH_VMCR_EL2.VFIQEn is read as one, write insensitive.

gicv3.cpuintf-mmap-access-level**Type**

int

Default value

0x0

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

The cpu supports separate trapping of ICC_DIR_EL1 to EL2.

gicv3.el3_trap_priority_when_secure_debug_disabled**Type**

bool

Default value

0x0

Undef to access priorities group register when secure debug is disabled.

gicv3.extended-interrupt-range-support**Type**

bool

Default value

0x0

Device has support for extended SPI/PPI ID ranges.

gicv3.gicv2-only**Type**

bool

Default value

0x0

Limit the GIC implementation to GICv2 features only.

gicv3.idle-is-ff**Type**

bool

Default value

0x1

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

Ignore UNPREDICTABLE access to GICC_DIR register.

gicv3.interrupt-bypass-support**Type**

bool

Default value

0x1

Interrupt bypass support, set to false for devices not supporting interrupt bypass.

gicv3.local-SEIs**Type**

bool

Default value

0x0

Generate SEI to signal internal issues.

gicv3.local-VSEIs**Type**

bool

Default value

0x0

Generate VSEI to signal internal issues.

gicv3.physical-ID-bits**Type**

int

Default value

0x10

Number of physical ID bits implemented.

gicv3.priority-bits**Type**

int

Default value

0x5

Number of priority bits implemented.

gicv3.send-PMHE-command-only-when-priority-changes**Type**

bool

Default value

0x0

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

Device has support for the Range Selector feature for SGI.

gicv3.suppress-virtual-enables-comms**Type**

bool

Default value

0x1

In GICv3 only mode, prevents the GIC CPUIF from communicating UpstreamWrite/VirtualEnables to the IRI.

gicv3.virtual-ID-bits**Type**

int

Default value

0x10

Number of virtual ID bits implemented.

gicv3.virtual-lpi-support**Type**

bool

Default value

0x1

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

Number of virtual priority bits implemented.

gicv3.without-DS-support**Type**

bool

Default value

0x0

GICv3 CPU interfaces do not support disabling security in the distributor (GICD_CTLR.DS=1).

gicv4.mask-virtual-interrupt**Type**

bool

Default value

0x0

If true, virtual interrupts can be masked from being reported to virtual CPU interface by setting ICH_HCR_EL2.DVIM 1. No control otherwise.

global_debug_rom.ROMDEVID**Type**

int

Default value

0x0

Value of Debug Rom Device Identification Register.

global_debug_rom.ROMPIDR**Type**

int

Default value

0x4000bb000

Value of Debug Rom Peripheral Identification Register.

global_debug_rom.ROMPRIDR0**Type**

int

Default value

0x1

Value of Debug ROM Power RequestID Register.

hardware_translation_table_update_implemented**Type**

int

Default value

0x1

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

GICv4.1 is enabled, and all the features with GICv4.1 are implemented (FEAT_GICv4p1).

has_16bit_asids**Type**

bool

Default value

0x1

Enable 16-bit ASIDs.

has_16bit_vmids**Type**

int

Default value

0x1

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

Implement the 16k LPAE translation granule.

has_4k_granule**Type**

bool

Default value

0x1

Implement the 4k LPAE translation granule.

has_64k_granule**Type**

bool

Default value

0x1

Implement the 64k LPAE translation granule.

has_aarch32_dbgdidr_etc**Type**

bool

Default value

0x1

DBGDIDR, DBGDRAR, DBGDSAR exist even if EL1 doesn't implement AArch32.

has_aarch64**Type**

bool

Default value

0x0

All implemented exception levels can run in AArch64.

has_bc**Type**

int

Default value

0x1

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

Implement the ARMv8R-64 CCSIDR Extension. Extending the ccidr number of sets (FEAT_CCIDX).

has_cluster_l1cache_size**Type**

bool

Default value

0x1

Whether core supports cluster level l1cache size.

has_coherent_icache**Type**

bool

Default value

0x0

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

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

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

If true, a debug ROM will be generated describing the cluster's debug components.

has_delayed_ctireg**Type**

bool

Default value

0x0

Delay the functional effect of CTI register writes until ISB or implicit barrier.

has_delayed_dbgreg**Type**

bool

Default value

0x0

Delay the functional effect of external debug register writes until ISB or implicit barrier.

has_delayed_mdscr_el1**Type**

bool

Default value

0x0

Delay the functional effect of MDSCR_EL1 register writes until ISB or implicit barrier.

has_delayed_oslar_el1**Type**

bool

Default value

0x0

Delay the functional effect of OSLAR_EL1 register writes until ISB or implicit barrier.

has_delayed_pmureg**Type**

bool

Default value

0x0

Delay the functional effect of PMU register writes until ISB or implicit barrier.

has_delayed_sysreg**Type**

bool

Default value

0x0

Delay the functional effect of system register writes until ISB or implicit barrier.

has_dgh**Type**

int

Default value

0x1

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

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

Implement EDACR register.

has_enhanced_pac**Type**

bool

Default value

0x0

If pointer authentication is enabled then implement enhanced PAC.

has_exception_trapping_form_of_vector_catch**Type**

bool

Default value

0x1

Implement the exception trapping form of vector catch debug event.

has_export_m_port**Type**

bool

Default value

0x1

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

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

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

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

Flash Port present.

has_flash_protection**Type**

bool

Default value

0x0

Implement flash memory protection.

has_fp16**Type**

int

Default value

0x1

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

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

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

Is Internal GIC IRI implemented.

`has_itd`

Type

bool

Default value

0x1

Implement the optional IT disable feature.

`has_large_system_ext`

Type

bool

Default value

0x0

Implement the ARMv8 Large System Extensions (FEAT_LSE).

`has_large_va`

Type

int

Default value

0x0

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_1lpp`

Type

bool

Default value

0x0

Low Latency Peripheral Port present.

has_mpm**Type**

bool

Default value

0x0

Implement max-power mitigation mechanism (MPMM).

has_no_os_double_lock**Type**

int

Default value

0x0

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

Whether PAR_EL1[10] is RAZ/WI.

has_partial_delayed_mdscr_el1**Type**

bool

Default value

0x0

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

If true, pc sample-based profiling is enabled (FEAT_PCSRv8, FEAT_PCSRv8p2).

has_per_cluster_debug_auth_ports**Type**

bool

Default value

0x0

If true then the debug authentication ports i.e. spniden, niden, rpliden, rtpiden, dbgen, spiden are available per cluster.

has_pl2**Type**

bool

Default value

0x1

Whether EL2 is implemented.

has_pmc**Type**

bool

Default value

0x0

Programmable MBIST controllers implemented.

has_pmu**Type**

int

Default value

0x1

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

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

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

Supports QARMA3 pointer authentication algorithm (FEAT_PACQARMA3).

has_ras**Type**

int

Default value

0x0

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

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

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

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

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

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

Implement software lock in memory-mapped CTI, PMU, and external debug interfaces.

has_speculation_barrier_inst**Type**

int

Default value

0x1

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

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

Shared Peripheral Port present.

has_stage2_ap_speculative_update**Type**

int

Default value

0x0

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

Report asynchronous abort due to unsupported load atomics as synchronous (Cacheable).

has_synchronous_load_atomics_noncacheable**Type**

bool

Default value

0x1

Report asynchronous abort due to unsupported load atomics as synchronous (Non-Cacheable).

has_synchronous_store_atomics**Type**

bool

Default value

0x0

Report asynchronous abort due to unsupported store atomics as synchronous (Cacheable).

has_synchronous_store_atomics_noncacheable**Type**

bool

Default value

0x0

Report asynchronous abort due to unsupported store atomics as synchronous (Non-Cacheable).

has_tlb_conflict_abort**Type**

bool

Default value

0x0

Detected inconsistent TLB content generate aborts.

has_tlb_pa_caching**Type**

bool

Default value

0x0

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

Report unsupported exclusive access with Unsupported Exclusive fault status (otherwise use external abort).

has_v8_4_pmu_extension**Type**

int

Default value

0x1

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

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

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

Implement write accesses buffering before L1 cache. May affect ext_abort behaviour.

hcptr_tta_behaviour**Type**

int

Default value

0x2

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

If true, HCR_EL2.MIOCNCE is treated as R/W instead of RAZ/WI.

hcr_swio_res1**Type**

bool

Default value

0x0

Whether HCR.SWIO and/or HCR_EL2.SWIO are RES1.

hsr_uncond_cc**Type**

bool

Default value

0x0

Condition codes reported in HSR as AL if it passes.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-prefetch_level`

Type

int

Default value

0x0

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

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

L1 I-Cache read bus width in bytes used to calculate per-access timing annotations.

`icache-read_latency`

Type

int

Default value

0x0

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

icache-ways**Type**

int

Default value

0x2

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

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

Behaviour of IMPLEMENTATION DEFINED registers and system instructions. 0, UNDEF. 1, RAZ/WI.

instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

internal_vgic**Type**

bool

Default value

0x0

Instantiate VGIC peripheral in this processor.

is_debug_state_pmu_snapshot_allowed**Type**

bool

Default value

0x1

If true, PMU snapshot is allowed in debug state.

is_first_pcsr_sample_ignored**Type**

bool

Default value

0x0

If true, First read of PMPCSR register after reset returns 0xFFFFFFFF.

is_serror_edge_triggered**Type**

bool

Default value

0x1

If true, SError is edge-triggered. Otherwise, its level-triggered.

is_uniprocessor**Type**

bool

Default value

0x0

Value for the U bit in MPIDR. true disables L1 cache coherency protocols.

isb_is_branch**Type**

bool

Default value

0x0

If true, ISB is traced as an immediate branch. This allows to count ISB as a branch in debug extensions (e.g. PMU).

ish_is_osh**Type**

bool

Default value

0x0

Whether Innershareable is same as OuterShareable.

itd_conditional_instructions_are_32bit**Type**

bool

Default value

0x0

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

If true, JIDR register access is UNDEF at EL0.

jmcr_is_undef_at_el0**Type**

bool

Default value

0x0

If true, JMCR register access is UNDEF at EL0.

joscr_is_undef_at_el0**Type**

bool

Default value

0x0

If true, JOSCR register access is UNDEF at EL0.

l1cache_has_r52_cache_policy**Type**

bool

Default value

0x0

Whether l1cache has r52 cache policy.

l1cache_has_rsvd_flash_ways**Type**

bool

Default value

0x0

Whether l1 cache segregates ways for flash and AXI data.

l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

l2cache-read_latency**Type**

int

Default value

0x0

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

L2 Cache size in bytes.

`l2cache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

`l2cache-write_access_latency`

Type

int

Default value

0x0

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

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

l2cache-write_latency**Type**

int

Default value

0x0

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

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

L3 Cache MPAMF_AIDR architecture major version.

l3cache-mpamf.arch_minor_ver**Type**

int

Default value

0x0

L3 Cache MPAMF_AIDR architecture minor version.

l3cache-mpamf.esr_mask**Type**

int

Default value

0xffffffff

L3 Cache MPAMF_ESR mask value.

l3cache-mpamf.has_esr**Type**

bool

Default value

0x0

L3 Cache's MPAMF_ESR, MPAMF_ECR, and MPAM error handling implemented.

l3cache-mpamf.has_extd_esr**Type**

bool

Default value

0x0

L3 Cache's MPAMF_ESR is 64-bits.

l3cache-mpamf.has_impl_idr**Type**

bool

Default value

0x0

L3 Cache's MPAMF_IMPL_IDR is present.

l3cache-mpamf.has_mbwu_long_counter**Type**

bool

Default value

0x0

L3 Cache has long MBWU counter and capture registers.

l3cache-mpamf.has_mpamfidr_ext**Type**

bool

Default value

0x0

MPAMF_IDR.EXT support.

l3cache-mpamf.has_partid_nrw**Type**

bool

Default value

0x0

Narrowing part ID register is present. This is global rather than per-instance.

l3cache-mpamf.has_priority_partitioning**Type**

bool

Default value

0x0

The selected resource has priority partitioning described in MPAMF_PRI_IDR.

l3cache-mpamf.has_prod_id**Type**

int

Default value

0x0

L3 Cache MPAMF_IIDR product ID supported.

l3cache-mpamf.has_prod_rev**Type**

int

Default value

0x0

L3 Cache MPAMF_IIDR product REVISION supported.

l3cache-mpamf.has_prod_var**Type**

int

Default value

0x0

L3 Cache MPAMF_IIDR product VARIENT supported.

l3cache-mpamf.has_ris**Type**

bool

Default value

0x0

L3 Cache has resource instance selection support.

l3cache-mpamf.max_partid_ns**Type**

int

Default value

0xffff

L3 Cache Maximum value of non-secure PARTID supported.

l3cache-mpamf.max_partid_rl**Type**

int

Default value

0xffff

L3 Cache Maximum value of realm PARTID supported for RME implementations.

l3cache-mpamf.max_partid_rt**Type**

int

Default value

0xffff

L3 Cache Maximum value of root PARTID supported for RME implementations.

l3cache-mpamf.max_partid_s**Type**

int

Default value

0xffff

L3 Cache Maximum value of secure PARTID supported.

l3cache-mpamf.max_pmg_ns**Type**

int

Default value

0xff

L3 Cache Maximum value of non-secure PMG supported.

l3cache-mpamf.max_pmg_rl**Type**

int

Default value

0xff

L3 Cache Maximum value of realm PMG supported for RME implementations.

l3cache-mpamf.max_pmg_rt**Type**

int

Default value

0xff

L3 Cache Maximum value of root PMG supported for RME implementations.

l3cache-mpamf.max_pmg_s**Type**

int

Default value

0xff

L3 Cache Maximum value of secure PMG supported.

l3cache-mpamf.mbwu_long_counter_width**Type**

int

Default value

0x0

L3 Cache long MBWU counter width. 0: 63 bits, 1: 44 bits.

l3cache-mpamf.no_impl_msmon**Type**

bool

Default value

0x0

L3 Cache's MPAMF_IMPL_IDR does not describe resource monitors.

l3cache-mpamf.no_impl_part**Type**

bool

Default value

0x0

L3 Cache's MPAMF_IMPL_IDR does not describe resource partitioning controls.

l3cache-mpamf.ris_max**Type**

int

Default value

0x0

L3 Cache's largest resource instance selector value defined.

l3cache-mpamf_base**Type**

int

Default value

0x0

L3 Cache memory mapped MPAM registers base address.

l3cache-read_access_latency**Type**

int

Default value

0x0

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

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

l3cache-read_latency**Type**

int

Default value

0x0

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

`l3cache-write_access_latency`

Type

int

Default value

0x0

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

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

`l3cache-write_latency`

Type

int

Default value

0x0

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

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

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**

bool

Default value

0x0

If true, FEAT_LS64 single-copy atomic 64-byte load/store instructions' memory attributes check is done on the combined memory attributes at the end of all enabled translation stages.

mdrar_el1_res0**Type**

bool

Default value

0x0

MDRAR_EL1 is RES0.

memory.ext_slave_base**Type**

int

Default value

0x0

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

Size of Slave region for each core.

`memory.flash_base`**Type**

int

Default value

0x0

Base address of Flash.

`memory.flash_size`**Type**

int

Default value

0x0

Size of the Flash RAM.

`memory.has_llram`**Type**

bool

Default value

0x0

Low-Latency RAM present.

`memory.l2_cache.is_inner_cacheable`**Type**

bool

Default value

0x1

L2 cache obeys inner cacheable attributes (rather than outer cacheable attributes).

`memory.l2_cache.is_inner_shareable`**Type**

bool

Default value

0x1

L2 cache obeys inner shareable attributes (rather than outer shareable attributes).

memory.llram_base**Type**

int

Default value

0x0

Base address of LLRAM.

memory.llram_enable_at_reset**Type**

bool

Default value

0x0

Whether llram is enabled at reset.

memory.llram_shared**Type**

bool

Default value

0x0

Controls the Low-Latency RAM's sharability attribute.

memory.llram_size**Type**

int

Default value

0x0

Size of the LLRAM.

memory.scu_present**Type**

bool

Default value

0x1

L1 Caches are coherent.

memory.transmit_vmid_in_user_flags**Type**

bool

Default value

0x0

Transmit VMID in transaction attributes.

mixed_endian**Type**

int

Default value

0x1

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

Layout of MPIDR. 0 AFF0 is CPUID, 1 AFF1 is CPUID.

non_secure_vgic_alias_when_ns_only**Type**

int

Default value

0x0

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

Number of v8-R protection regions.

num_protection_regions_s2**Type**

int

Default value

0x20

Number of v8-R hyp protection regions.

num_spi**Type**

int

Default value

0x20

Number of interrupts (SPI) into the internal GIC controller.

number_of_error_records**Type**

int

Default value

0x0

Cores Number of Error records supported for RAS.

page_based_hardware_attributes**Type**

int

Default value

0x0

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

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.

per_core_master_supported**Type**

bool

Default value

0x0

If master port from each core is exposed out of cluster.

pfr1_csv2_frac**Type**

int

Default value

0x0

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

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

Defines whether the address translation performed by the Profiling Buffer manages the Access Flag and dirty state.

pmcr_disable_events_export**Type**

bool

Default value

0x1

If true, export for PMU events is disabled. This configures PMCFGR.EX field.

pme-cr-res0-fields-razwi**Type**

bool

Default value

0x0

If true, RES0 fields of PME-CR_EL1 are accessed with a Read-As-Zero/Write-Ignore policy.

pmmir-el1-bus-slots**Type**

int

Default value

0x0

Largest value by which BUS_ACCESS can increment over BUS_CYCLES cycles.

pmmir-el1-bus-width**Type**

int

Default value

0x0

Width, in bytes, of accesses counted by BUS_ACCESS.

pms-idr-max-size**Type**

int

Default value

0x6

Defines largest size for a single SPE record (rounded up to a power of 2).

pmu-num-counters**Type**

int

Default value

0x8

Number of PMU counters implemented.

pmu_cycle_counter_counts_actual_cycles**Type**

bool

Default value

0x0

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

PMU (if present) implements event number 0x1e, CHAIN.

pseudo_fault_generation_feature_register**Type**

string

Default value

""

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

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

ram_protection_enable_at_reset**Type**

bool

Default value

0x0

Enable TCM, L1Cache memory protection after reset.

randomize_unknowns_at_reset**Type**

bool

Default value

0x0

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

""

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

0x18

RAS Pseudo-Fault generation clock rate in MHz.

register_reset_data**Type**

int

Default value

0x0

Data used to fill register bits when they become UNKNOWN at reset.

register_reset_data_hi**Type**

int

Default value

0x0

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

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

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

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

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

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

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

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

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

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).

rnr_always_implemented**Type**

bool

Default value

0x0

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

If true, s1 alignment fault has priority over s2 permission faults.

scheduler_mode**Type**

int

Default value

0x0

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

Whether SCR.nET is writeable. Writing to it is purely cosmetic (nET behavior not implemented).

scramble_unknowns_at_reset**Type**

bool

Default value

0x1

Will fill in unknown bits in registers at reset with register_reset_data.

serror_clear_delay**Type**

int

Default value

0x0

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

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

Sets the base address of Shared Peripheral Port.

spp.size**Type**

int

Default value

0x1000

Sets the size of SPP(in bytes).

spsr_el3_is_mapped_to_spsr_mon**Type**

bool

Default value

0x0

Whether SPSR_EL3 is mapped to AArch32 register SPSR_mon.

spsr_m4_res0**Type**

bool

Default value

0x0

Whether SPSR_ELx.M[4] bit should be RES0 for AARCH64 only implementations.

stage12_tlb_size**Type**

int

Default value

0x0

If VMSEA 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

Number of stage1 only tlb entries.

stage1_walkcache_size**Type**

int

Default value

0x0

Number of stage1 only walk cache entries.

strex_fail_can_hit_watchpoint**Type**

bool

Default value

0x0

If true, a strex fail can hit watchpoint.

supports_multi_threading**Type**

bool

Default value

0x0

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

If true, swp with dest as xzr is treated as store atomic.

take_ccfail_tsc_trap**Type**

bool

Default value

0x0

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

UNDEF exception is taken even if condition code check fails.

tidcp_traps_el0_undef_imp_def**Type**

bool

Default value

0x1

TIDCP has priority over UNDEF for accesses to IMPLEMENTATION DEFINED functionality from EL0.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_or_ic_invalid_xt**Type**

int

Default value

0x0

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

ETM trace registers support access via system registers.

trace_icc_registers_as_icv_when_redirected**Type**

bool

Default value

0x0

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

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).

trap_dc_cmo_to_pou_if_nop**Type**

bool

Default value

0x1

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

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

Whether setting HCR_EL2.TID3 traps reserved group3 id registers.

treat-dcache-cmos-to-poc-as-nop**Type**

int

Default value

0x0

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

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

Treat data cache invalidate operations as clean and invalidate.

treat-icache-cmos-to-pou-as-nop**Type**

int

Default value

0x0

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

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

If true, treat PLD as NOP.

treat_pli_as_nop**Type**

bool

Default value

0x0

If true, treat PLI as NOP.

treat_wfi_wfe_as_nop**Type**

bool

Default value

0x0

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

On Illegal ERET to AArch32, truncate PC to 32-bits.

unification-level**Type**

int

Default value

0x1

Level of Unification Inner Shareable for the cache hierarchy.

unification-uniprocessor-level**Type**

int

Default value

0x1

Level of Unification Uniprocessor for the cache hierarchy.

unpred_LSE128_overlap**Type**

int

Default value

0x1

Constrained unpredictable behaviours for 128-bit LSE overlap. 1 Constraint_UNDEF, 2 Constraint_NOP.

unpred_clear_ISV_for_exception_before_software_step**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

HVC unpredictable behaviour. 0, UNDEF. 1, NOP.

unpredictable_smc_behaviour**Type**

int

Default value

0x0

SMC unpredictable behaviour. 0, UNDEF. 1, NOP.

unpredictable_wfet_and_wfit_behaviour**Type**

int

Default value

0x1

WFET and WFIT unpredictable behaviour in debug state. 0, UNDEFINED. 1, NOP.

unsupported_atomic_fault_type**Type**

int

Default value

0x0

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

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

Use names SP/LR/PC instead of R13/R14/R15.

use_stage1_sh_as_input_to_stage2**Type**

bool

Default value

0x0

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

Translation table entries with the contiguous hint bit set generate large TLB entries.

user_defined_rom_table_debug_power_config**Type**

string

Default value

""

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

VPU data path width.

walk_cache_latency**Type**

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

warn_unpredictable_in_v7**Type**

bool

Default value

0x1

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

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

Configure WFE wakeup delay in CPU cycles.

wfi_wakeup_delay**Type**

int

Default value

0x0

Configure WFI wakeup delay in CPU cycles.

wp_ignores_dbm_update

Type

bool

Default value

0x0

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 3-191: 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 23.

Changes in 11.27.17

Ports added:

- pmusnapshotacks
- pmusnapshotreqs

Parameters added:

- has_arm_v9-6
- has_cmpbr
- has_f8f16mm
- has_f8f32mm
- has_feat_pops
- has_fprcvt
- has_lsfe
- has_lsui
- has_occmo
- has_pcdphint
- has_pmu_extpmn

- `has_ras_mmap_view`
- `has_rme_gdi`
- `has_rme_gpc3`
- `has_spe_nvm`
- `has_srmask`
- `has_uinj`
- `has_v9_6_spe_extension`
- `illegal_state_exception_priority`
- `l3cache-mpamf.esr_mask`
- `mpam_bw_bwa_wd`
- `mpam_bw_has_hw_scale`
- `mpam_bw_max_lim`
- `mpam_bw_us_frac`
- `mpam_bw_us_int`
- `mpam_has_bw_ctrl`
- `nv_frac_support_level`
- `pmecr_res0_fields_razwi`
- `poison_range_end_addr`
- `poison_range_start_addr`
- `pstate_pm_reset`
- `ras_mmap_address`
- `rme_nsh_cacheable_is_shareable`
- `sve.has_bfscale`
- `sve.has_sme_b16b16`
- `sve.has_ssve_aes`
- `sve.has_ssve_f8f16mm`
- `sve.has_ssve_f8f32mm`
- `sve.has_sve_f16f32mm`
- `sve.sme_highest_implemented_priority`
- `treat-dcache-cmos-to-occ-as-nop`

Iris and MTI instances for AEMvACT

This model has the following Iris instances:

Table 3-192: AEMvACT Iris instances

InstanceName	ComponentName
AEMvACT	Cluster_ARM_AEM-A_MP
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.dtlb	TlbCadi
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.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

InstanceName	ComponentName
AEMvACT.l2_cache.upstream[8]	PVBusSlave
AEMvACT.l2_cache.upstream[9]	PVBusSlave
AEMvACT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-193: AEMvACT MTI instances

InstanceName	ComponentName
AEMvACT	ARMv8Cluster
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

InstanceName	ComponentName
AEMvACT.12_cache.upstream[3]	PVBusSlave
AEMvACT.12_cache.upstream[4]	PVBusSlave
AEMvACT.12_cache.upstream[5]	PVBusSlave
AEMvACT.12_cache.upstream[6]	PVBusSlave
AEMvACT.12_cache.upstream[7]	PVBusSlave
AEMvACT.12_cache.upstream[8]	PVBusSlave
AEMvACT.12_cache.upstream[9]	PVBusSlave
AEMvACT.12_flusher	AsyncCacheFlushUnit

AEMvACT contains the following CADI targets:

- ARM_AEM-A_MP
- Cluster_ARM_AEM-A_MP
- PVCache
- TlbCadi

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 3-194: Architectural features implemented in Fast Models

Architecture version	Feature	Fast Models version									
		11.27	11.26	11.25	11.24	11.23	11.22	11.21	11.20	11.19	11.18
Armv8.3-A	All. For details, see list after table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Armv8.4-A	All. For details, see list after table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Architecture version	Feature	Fast Models version									
		11.27	11.26	11.25	11.24	11.23	11.22	11.21	11.20	11.19	11.18
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	✓	✓	✓	✓	✓	✓	✓	✓	-	-
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.	✓	✓	✓	✓	✓	✓	✓	-	-	-

Architecture version	Feature	Fast Models version									
		11.27	11.26	11.25	11.24	11.23	11.22	11.21	11.20	11.19	11.18
	All other features.	✓	✓	✓	✓	✓	✓	✓	✓	-	-
Armv9.5-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)
- EnhancedPAC2
- 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.

Ports for AEMvACT

Table 3-195: 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 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 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[8]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[8]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[8]	Signal	Master	Timer signals to SOC.
CNTHVSIQ[8]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[8]	Signal	Master	Timer signals to SOC.
CNTPSIQ[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.
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.

Name	Protocol	Type	Description
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
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	-

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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

ADFSR and AIFSR are implemented.

AIDR

Type

int

Default value

0x0

Value of AIDR_EL1 register.

AMIIDR

Type

int

Default value

0x43b

Value of AMU Implementation Identification Register.

AMPIDR**Type**

int

Default value

0x4000bb000

Value of AMU Peripheral Identification Register.

BPIMVA_causes_translation_lookup**Type**

bool

Default value

0x0

Do a translation when BPIMVA instruction is executed (which may cause a translation fault).

BROADCASTCACHEMAINT**Type**

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTINNER**Type**

bool

Default value

0x1

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CCSIDR-L1D_override**Type**

int

Default value

0x0

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

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

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

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Value of CPU Configuration Register.

CTIPIDR**Type**

int

Default value

0x0

If non-zero, override the CTI Peripheral Identification Register.

CTR-L1Ip-override**Type**

int

Default value

0x0

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

If true, RES0 behaviour is applied to DBGBCR(_EL1).BT before checking for reserved values for this field.

DBGPIDR**Type**

int

Default value

0x0

If non-zero, override the Debug Peripheral Identification Register.

DBGROMADDR**Type**

int

Default value

0x0

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x0

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

ERRIIDR**Type**

int

Default value

0xd800143b

Value of RAS Implementation Identification Register.

ERRPIDR**Type**

int

Default value

0x4100bbd80

Value of RAS Peripheral Identification Register.

ERXMISC0_mask**Type**

int

Default value

0x0

Write Mask for ERXMISC0 RAS Register.

GICDISABLE**Type**

bool

Default value

0x1

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

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

Whether ESR_EL2.ISV is set to 0 on stage 2 synchronous external aborts.

MIDR**Type**

int

Default value

0x410fd0f0

Value of MIDR_EL1 register.

NUM_CORES**Type**

int

Default value

0x1

Number of cores per cluster.

PA_SIZE**Type**

int

Default value

0x28

Physical address range supported For ARMv8.0 and ARMv8.1 this is limited to 48 bits (FEAT_LPA/FEAT_LPA3).

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

PMCEID0**Type**

int

Default value

0xffffffff

Performance Monitor Common Event ID Reg 0 value - 64 bit.

PMCEID1**Type**

int

Default value

0xffffffff

Performance Monitor Common Event ID Reg 1 value - 64 bit.

PMSIDR.ArchInst**Type**

bool

Default value

0x1

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

Defines whether call return branch records (FEAT_SPE_CRR) is implemented or not.

PMSIDR.LDS**Type**

bool

Default value

0x0

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

If non-zero, override the PMU Peripheral Identification Register.

abort_execution_from_device_memory**Type**

bool

Default value

0x0

Execution from device memory generates a prefetch abort.

advsimd_bf16_support_level**Type**

int

Default value

0x0

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

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

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

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

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

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

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

RW mask for implementation-defined registers.

amu_aux_type_fixed**Type**

string

Default value

""

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

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

Implement external memory-mapped access to system register of activity monitor unit from ARMv8.4. 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

Implement system register access to activity monitor unit from ARMv8.4.

amu_mmap_address**Type**

string

Default value

""

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

Number of AMU auxiliary counters implemented.

amu_reset_domain**Type**

int

Default value

0x0

Reset domain for activity monitor unit. 0, COLD_RESET. 1, WARM_RESET. 2, NONE.

amu_version**Type**

int

Default value

0x1

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

At EL0, unknown bits of APSR are RAZ.

arm_v8_7_accelerator_support_level**Type**

int

Default value

0x0

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_priority**Type**

int

Default value

0x0

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

Value of AFRO ID register.

branch-predictor-clear-policy

Type

int

Default value

0x2

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

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_log2_num_records

Type

int

Default value

0x6

Log2 of number of BRB records supported. 3 -> 8 records, ... 6 -> 64 records.

cache-log2linelen

Type

int

Default value

0x6

Log2 of the cache line length in bytes.

cache_maintenance_hits_watchpoints

Type

bool

Default value

0x0

DCIMVA operations executed in AArch32 modes hit watchpoints.

changing_block_size_without_bbm_support

Type

int

Default value

0x0

Level of support for changing block size without break-before-make (FEAT_BBM).

check_memory_attributes

Type

bool

Default value

0x1

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

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

Clean and invalidate caches on warm reset.

clear_IT_when_IL_set**Type**

bool

Default value

0x0

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

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

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

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

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

""

"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

""

"[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

""

"[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

""

"[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

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

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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CONFIG64**Type**

bool

Default value

0x1

Register width configuration at reset. 0, AArch32. 1, AArch64.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.CP15SDISABLE2**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers (FEAT_CP15SDISABLE2).

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.DCZID-log2-block-size**Type**

int

Default value

0x8

Log2 of the block size cleared by DC ZVA instruction (as read from DCZID_ELO).

cpuX.DCZVA_single_write**Type**

bool

Default value

0x0

Execute the DCZVA as a single write.

cpuX.MPIDR-override**Type**

int

Default value

0x0

Override of MPIDR value. If nonzero will override the MT, cluster and CPU ID bits in MPIDR.

cpuX.RVBAR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.RVBAR32**Type**

int

Default value

0x0

Reset vector address in AARCH32 when VINITHI is not set and ignore_rvbar_in_aarch32 is set.

cpuX.SMPnAMP**Type**

bool

Default value

0x1

Enable broadcast messages necessary for correct SMP operation at reset.

cpuX.TEINIT**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.aarch32_reset_from_impdef_addr**Type**

bool

Default value

0x1

If PE resets into AArch32, Whether execution starts from IMPDEF address or hi/low vector.

cpuX.ase-present**Type**

bool

Default value

0x1

Set whether the model has been built with NEON support.

cpuX.clock_divider**Type**

int

Default value

0x1

Clock divider ratio for asymmetric MP clocking.

cpuX.clock_multiplier**Type**

int

Default value

0x1

Clock divider ratio for asymmetric MP clocking.

cpuX.crypto_aes**Type**

int

Default value

0x2

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

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

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

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

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

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

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

Set bits represent that the corresponding trigger requires software acknowledge via CTIINTACK.

cpuX.cti-number_of_claim_bits**Type**

int

Default value

0x0

Number of implemented bits in CTICLAIMSET.

cpuX.cti-number_of_triggers**Type**

int

Default value

0x8

Number of cti event triggers (default: 8, valid values: {3, 8-32}).

cpuX.enable_crc32**Type**

int

Default value

0x0

CRC32 instructions supported. 0, not implemented. 1, CRC32 instructions implemented (FEAT_CRC32).

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.etm-present**Type**

bool

Default value

0x1

Set whether the model has ETM support.

cpuX.force-fpsid**Type**

bool

Default value

0x0

Override the FPSID value.

cpuX.force-fpsid-value**Type**

int

Default value

0x0

Value to override the FPSID value to.

cpuX.has_hcptr_tase**Type**

bool

Default value

0x1

If false, HCPTR.TASE is RES0.

cpuX.max_code_cache_mb**Type**

int

Default value

0x100

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.number-of-breakpoints**Type**

int

Default value

0x10

Number of breakpoints.

cpuX.number-of-context-breakpoints**Type**

int

Default value

0x10

Number of breakpoints that are context aware.

cpuX.number-of-watchpoints**Type**

int

Default value

0x10

Number of watchpoints.

cpuX.operation_bandwidth**Type**

int

Default value

0x1

Operation width for ARMv8.4 PMU extension.

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-prefix**Type**

bool

Default value

0x0

Prefix semihosting output with target instance name.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.semihosting-stderr_istty**Type**

bool

Default value

0x1

Result for semihost istty call when argument is stderr.

cpuX.semihosting-stdin_istty**Type**

bool

Default value

0x1

Result for semihost istty call when argument is stdin.

cpuX.semihosting-stdout_istty**Type**

bool

Default value

0x1

Result for semihost istty call when argument is stdout.

cpuX.semihosting-use_stderr**Type**

bool

Default value

0x0

Send stderr from the simulated process to host stderr.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

Set whether the model has VFP support.

cpuX.vfp-traps**Type**

bool

Default value

0x1

Implement support for trapping floating-point exceptions.

cpuX.vfp-traps-show-all**Type**

bool

Default value

0x0

Report all trapped floating-point exceptions in the syndrome when a combination occurs.

cpuX.wfet_early_or_delayed_timeout**Type**

int

Default value

0x0

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

WFIT early or delayed timeout beyond the threshold value of CNTVCT_ELO in percentage.

cpuselr_el3_sync_immediate**Type**

bool

Default value

0x1

Adjust when the patching selection register synchronises - either immediately (true - default), or awaiting for barrier event.

cpy_mops_option**Type**

int

Default value

0x0

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

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

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

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

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

Debug ROM dap base address.

dbgitr_buffer_size

Type

int

Default value

0x0

Number of instructions which can be buffered before EDSCR.ITE is cleared.

dbgxvr_ress_is_stateful

Type

bool

Default value

0x0

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

`dcache-read_access_latency`

Type

int

Default value

0x0

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

L1 D-Cache read bus width in bytes used to calculate per-access timing annotations.

dcache-read_latency**Type**

int

Default value

0x0

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-ways**Type**

int

Default value

0x2

L1 D-Cache number of ways (sets are implicit from size).

dcache-write_access_latency**Type**

int

Default value

0x0

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

L1 D-Cache write bus width in bytes used to calculate per-access timing annotations.

dcache-write_latency**Type**

int

Default value

0x0

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

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

Whether DC ZVA reports lowest address in FAR on tag check fail.

dczva_wp_far_behaviour

Type

int

Default value

0x0

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

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

""

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

""

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

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

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

If true, present a debug ROM table recommended by ARMv8 Debug Architecture. Otherwise, use nested ROM tables.

delay_serror**Type**

int

Default value

0x0

Add a propagation delay of serror signal into the core.

delayed_dbgreg_between_secure_views**Type**

int

Default value

0x1

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

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

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

Number of shared peripheral interrupts implemented.

disable_sve_plugin**Type**

bool

Default value

0x0

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

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

Limit the maximum number of observable DSB side effects which can be queued (e.g. TLBI), after which DSB sync will be done automatically.

early_implicit_error_sync_event_behaviour**Type**

int

Default value

0x0

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

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

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

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

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

If true then UNKNOWN bits in RAS field in EDPFR are read as 0.

edxfr_reads_idreg**Type**

bool

Default value

0x0

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 'eddfrr*_reads_idreg_mask' parameter.

el0_can_access_imp_def_functionality**Type**

bool

Default value

0x0

If not made UNDEF by imp_def_functionality_behaviour, EL0 can access IMPLEMENTATION DEFINED registers and system instructions.

el0_el1_only_non_secure**Type**

bool

Default value

0x0

Secure/non-secure state if EL2 and EL3 are not implemented. 0, secure. 1, non-secure.

el3_trap_priority_when_secure_debug_disabled**Type**

bool

Default value

0x0

Undef when secure debug is disabled (EDSCR.SDD == 1) && boolean IMPLEMENTATION_DEFINED 'EL3 trap priority when SDD == 1'.

enable_address_contig_check**Type**

bool

Default value

0x0

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

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

Perform extra pagetable walks to check translation table entries that have the contiguous hint bit set.

enhanced_pac2_level**Type**

int

Default value

0x0

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

""

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

""

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

Mask for stateful bits for ERXPFGCTL which are RES0.

esr_write_update_res0

Type

bool

Default value

0x0

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

Generate the non-periodic alignment synchronisation packet generation when trace unit is operative.

ete.ATBTRIG

Type

bool

Default value

0x1

ATB trigger support.

ete.CCITMIN**Type**

int

Default value

0x4

Minimum cycle count value.

ete.CCSIZE**Type**

int

Default value

0xc

Cycle counter size.

ete.CLAIMTAGS**Type**

int

Default value

0x8

Number of claim tags.

ete.COMMOPT**Type**

bool

Default value

0x1

Commit mode.

ete.COMMTRANS**Type**

bool

Default value

0x0

Commit transaction mode.

ete.DEBUG**Type**

int

Default value

0x2

DEBUG.

ete.DESIGNER**Type**

int

Default value

0x41

DESIGNER value.

ete.ETE_REVISION**Type**

int

Default value

0x0

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

ete.EXCEPTION_WITH_CONTEXT**Type**

bool

Default value

0x1

Whether EXCEPTION_WITH_CONTEXT packet is supported.

ete.EXPLICITLY_COMMIT_PO_ELEMS**Type**

bool

Default value

0x0

Whether to unilaterally explicitly emit a commit after a PO packet.

ete.IMPDEFEXCEPPERCENTAGE**Type**

int

Default value

0x0

Percentage of IMPDEF exceptions inserted in instruction blocks.

ete.IMPDEF_TRACE_ON**Type**

int

Default value

0x0

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

Number of instruction blocks traced on a transition in the filtering.

ete.LPOVERRIDE**Type**

bool

Default value

0x1

Low power override.

ete.MAXSPEC**Type**

int

Default value

0x0

Maximum speculation depth.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NOOVERFLOW**Type**

bool

Default value

0x0

No overflow.

ete.NUMACPAIRS**Type**

int

Default value

0x4

Number of instruction address comparators pairs.

ete.NUMCIDC**Type**

int

Default value

0x1

Number of context ID comparators.

ete.NUMCNTR**Type**

int

Default value

0x2

Number of counters.

ete.NUMEXTINSEL**Type**

int

Default value

0x4

Number of external input selectors.

ete.NUMPC**Type**

int

Default value

0x0

Number of PE comparators.

ete.NUMSEQSTATE**Type**

int

Default value

0x4

Number of sequencer states.

ete.NUMSSCC**Type**

int

Default value

0x1

Number of single shot comparators.

ete.NUMVMIDC**Type**

int

Default value

0x1

Number of virtual ID comparators.

ete.NumberOfETEEEvents**Type**

int

Default value

0x2

Number of trace events.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.PIDR_DESIGNER**Type**

int

Default value

0x0

TRCPIDR DESIGNER value.

ete.PIDR_PART**Type**

int

Default value

0x0

TRCPIDR PART number value.

ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.QFILT**Type**

bool

Default value

0x0

Q filtering.

ete.QSUP**Type**

int

Default value

0x0

Q support.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RAZWI_REG_SEL_TOP_BIT**Type**

bool

Default value

0x0

Implement Resource Selectors or Resource Selector Pairs bits as RAZ/WI.

ete.REG_ACCESS_ONLY_MODE**Type**

bool

Default value

0x0

If enabled, all traces are disabled. Plugin only allows register acceses.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.STALLCTRL**Type**

bool

Default value

0x1

Stall control.

ete.SYSSTALL**Type**

bool

Default value

0x1

System stall.

ete.TRACEIDSIZE**Type**

int

Default value

0x7

Trace ID size.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRACE_OUTPUT_ENABLE**Type**

bool

Default value

0x0

ETE Trace output enable: 1=enable, 0=disable.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

ete.TSMARK**Type**

bool

Default value

0x0

Whether timestamp markers are supported.

ete.TSSIZE**Type**

int

Default value

0x8

Timestamp size.

ete.WFXMODE**Type**

bool

Default value

0x1

WFX mode.

ets_level**Type**

int

Default value

0x0

Level of Enhanced Translation Synchronization supported. 0: not supported (may be override to 1 if has_ets=true), 1: supported FEAT_ETS, 2: supported FEAT_ETS2.

exception_catch_before_software_step**Type**

bool

Default value

0x1

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

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

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

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

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

Exclusive monitors in the cluster will be cleared when a strex succeeds.

exercise_stxr_fail**Type**

bool

Default value

0x0

Reject a pseudo-random majority of exclusive store instructions.

ext_abort_device_GRE_prefetch_ras_index**Type**

int

Default value

0xffffffffffffffff

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

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

Critical reporting of device-GRE read external aborts.

ext_abort_device_GRE_read_is_sync**Type**

int

Default value

0x2

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

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

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

Critical reporting of device-GRE write external aborts.

ext_abort_device_GRE_write_is_sync**Type**

int

Default value

0x2

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

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

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

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

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

Critical reporting of device-nGRE read external aborts.

ext_abort_device_nGRE_read_is_sync**Type**

int

Default value

0x2

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

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

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

Critical reporting of device-nGRE write external aborts.

ext_abort_device_nGRE_write_is_sync**Type**

int

Default value

0x2

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

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

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

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

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

Synchronous reporting of device read with acquire external aborts.

ext_abort_device_read_is_critical**Type**

bool

Default value

0x0

Critical reporting of device-nGnRE read external aborts.

ext_abort_device_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_read_ras_index**Type**

int

Default value

0x0

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

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

Critical reporting of device-nGnRE write external aborts.

ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_device_write_ras_index**Type**

int

Default value

0x0

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

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

Returned data, if external aborts are asynchronous.

ext_abort_normal_cacheable_read_is_critical**Type**

bool

Default value

0x0

Critical reporting of normal write-back cacheable-read external aborts.

ext_abort_normal_cacheable_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of normal write-back cacheable-read external aborts.

ext_abort_normal_cacheable_read_ras_index**Type**

int

Default value

0x0

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

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

Critical reporting of normal write-back cacheable write external aborts.

ext_abort_normal_cacheable_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of normal write-back cacheable write external aborts.

ext_abort_normal_cacheable_write_ras_index**Type**

int

Default value

0x0

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

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

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

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

Critical reporting of normal noncacheable-read external aborts.

ext_abort_normal_noncacheable_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of normal noncacheable-read external aborts.

ext_abort_normal_noncacheable_read_ras_index**Type**

int

Default value

0x0

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

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

Critical reporting of normal noncacheable write external aborts.

ext_abort_normal_noncacheable_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of normal noncacheable write external aborts.

ext_abort_normal_noncacheable_write_ras_index**Type**

int

Default value

0x0

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

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

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

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

Critical reporting of normal write-through cacheable-read external aborts.

ext_abort_normal_wt_cacheable_read_is_sync**Type**

int

Default value

0x2

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

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

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

Critical reporting of normal write-through write external aborts.

ext_abort_normal_wt_cacheable_write_is_sync**Type**

int

Default value

0x2

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

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

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

Critical reporting of external aborts generated by device-GRE instruction fetches.

ext_abort_prefetch_device_GRE_read_is_sync

Type

int

Default value

0x2

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

Critical reporting of external aborts generated by device-nGRE instruction fetches.

ext_abort_prefetch_device_nGRE_read_is_sync

Type

int

Default value

0x2

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

Critical reporting of external aborts generated by device-nGnRE instruction fetches.

ext_abort_prefetch_device_read_is_sync**Type**

int

Default value

0x2

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

Critical reporting of external aborts generated by normal writeback cacheable instruction fetches.

ext_abort_prefetch_is_sync**Type**

bool

Default value

0x1

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

Critical reporting of external aborts generated by normal noncacheable instruction fetches.

ext_abort_prefetch_noncacheable_read_is_sync**Type**

int

Default value

0x2

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

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

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

Critical reporting of external aborts generated by device-nGnRnE instruction fetches.

`ext_abort_prefetch_so_read_is_sync`

Type

int

Default value

0x2

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

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

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

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

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

Critical reporting of device-nGnRnE read external aborts.

ext_abort_so_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_read_ras_index**Type**

int

Default value

0x0

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

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

Critical reporting of device-nGnRnE write external aborts.

ext_abort_so_write_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of device-nGnRnE write external aborts.

ext_abort_so_write_ras_index**Type**

int

Default value

0x0

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

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

Critical reporting of TTW cacheable read external aborts.

ext_abort_ttw_cacheable_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of TTW cacheable read external aborts.

ext_abort_ttw_cacheable_read_ras_index**Type**

int

Default value

0x0

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

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

Critical reporting of TTW noncacheable read external aborts.

ext_abort_ttw_noncacheable_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of TTW noncacheable read external aborts.

ext_abort_ttw_noncacheable_read_ras_index**Type**

int

Default value

0x0

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

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

Critical reporting of TTW write-through cacheable read external aborts.

ext_abort_ttw_wt_cacheable_read_is_sync**Type**

int

Default value

0x2

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

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

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

Configure External Debug Request delay in CPU cycles.

external_oslar_access_disabled_by_authentication**Type**

bool

Default value

0x0

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

Whether GPF faults occur when GPT contiguous entries are misprogrammed.

fault_on_nT_bit_set**Type**

bool

Default value

0x1

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

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

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

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

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

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_wnr_read_unsupported_exclusive_or_atomic**Type**

bool

Default value

0x0

Whether ESR_ELx.WnR is forced to 0 for unsupported atomic and exclusives.

force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

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 MPAMO_EL1, MPAM1_EL1, MPAM2_EL2, MPAM3_EL3, MPAMHCR_EL2.

fp8_support_level**Type**

int

Default value

0x0

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

FPSCR and FPCR fields LEN and STRIDE are hardwired to 0.

fpsr_res0_stateful_mask**Type**

int

Default value

0x0

Mask for stateful bits of FPSR which are RES0.

fsr_ext_bit_update_kind**Type**

int

Default value

0x3

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

GCS Data check exceptions are taken before Data Aborts.

gcs_overshoot_writes**Type**

int

Default value

0x0

Number of overshooting GCS records written after a writing a record.

gic.GICC-offset**Type**

int

Default value

0x2000

Offset from PERIPHBASE for GICC registers.

gic.GICD-offset**Type**

int

Default value

0x1000

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

Offset from PERIPHBASE for GICH registers.

gic.GICH-other-CPU-offset**Type**

int

Default value

0x5000

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

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

Offset from PERIPHBASE for GICV registers.

gic.PERIPH-size**Type**

int

Default value

0x8000

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

Whether a non-zero value for affinity at level 3 is supported.

`gicv3.BPR-min`**Type**

int

Default value

0x2

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

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

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

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

Ignore EOI writes that cannot end the highest priority active interrupt.

`gicv3.FIQEn-RAO`**Type**

bool

Default value

0x0

GICC_CTLR.FIQEn is read as one, write insensitive.

`gicv3.IIDR_base`**Type**

int

Default value

0x43b

The base value for calculating the GICC_IIDR register value.

`gicv3.LR-count`**Type**

int

Default value

0x10

The number of implemented list registers.

`gicv3.PMHE-RAO-WI`**Type**

bool

Default value

0x0

ICC_CTLR_EL*.PHME is read as one, write insensitive.

`gicv3.PMHE-RAZ-WI`**Type**

bool

Default value

0x0

ICC_CTLR_EL*.PHME is read as zero, write insensitive.

gicv3.PMHE-release-set-packet

Type

bool

Default value

0x0

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

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

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

Restrict SRE EL3 to be set only once.

gicv3.SRE-enable-action-on-mmap

Type

int

Default value

0x0

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

If GICv3 CPU interface is being used, this determines whether the STATUS registers are implemented.

gicv3.VBPR-min**Type**

int

Default value

0x2

The minimum value for the GICV_BPR register (non-secure version will be 1 + this value).

gicv3.VFIQEn-RAO**Type**

bool

Default value

0x0

ICH_VMCR_EL2.VFIQEn is read as one, write insensitive.

gicv3.cpuintf-mmmap-access-level**Type**

int

Default value

0x0

Allowed values are: 0-mmmap access is supported for GICC,GICH,GICV registers. 1-mmmap access is supported only for GICV registers. 2-mmmap access is not supported.

gicv3.dir-trap-support**Type**

bool

Default value

0x1

The cpu supports separate trapping of ICC_DIR_EL1 to EL2.

gicv3.el3_trap_priority_when_secure_debug_disabled**Type**

bool

Default value

0x0

Undef to access priorities group register when secure debug is disabled.

gicv3.extended-interrupt-range-support**Type**

bool

Default value

0x0

Device has support for extended SPI/PPI ID ranges.

gicv3.gicv2-only**Type**

bool

Default value

0x0

Limit the GIC implementation to GICv2 features only.

gicv3.idle-is-ff**Type**

bool

Default value

0x1

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

Ignore UNPREDICTABLE access to GICC_DIR register.

gicv3.interrupt-bypass-support**Type**

bool

Default value

0x1

Interrupt bypass support, set to false for devices not supporting interrupt bypass.

gicv3.local-SEIs**Type**

bool

Default value

0x0

Generate SEI to signal internal issues.

gicv3.local-VSEIs**Type**

bool

Default value

0x0

Generate VSEI to signal internal issues.

gicv3.physical-ID-bits**Type**

int

Default value

0x10

Number of physical ID bits implemented.

gicv3.priority-bits**Type**

int

Default value

0x5

Number of priority bits implemented.

gicv3.send-PMHE-command-only-when-priority-changes**Type**

bool

Default value

0x0

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

Device has support for the Range Selector feature for SGI.

gicv3.suppress-virtual-enables-comms**Type**

bool

Default value

0x1

In GICv3 only mode, prevents the GIC CPUIF from communicating UpstreamWrite/VirtualEnables to the IRI.

gicv3.virtual-ID-bits**Type**

int

Default value

0x10

Number of virtual ID bits implemented.

gicv3.virtual-lpi-support**Type**

bool

Default value

0x1

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

Number of virtual priority bits implemented.

gicv3.without-DS-support**Type**

bool

Default value

0x0

GICv3 CPU interfaces do not support disabling security in the distributor (GICD_CTLR.DS=1).

gicv4.mask-virtual-interrupt**Type**

bool

Default value

0x0

If true, virtual interrupts can be masked from being reported to virtual CPU interface by setting ICH_HCR_EL2.DVIM 1. No control otherwise.

global_debug_rom.ROMDEVID**Type**

int

Default value

0x0

Value of Debug Rom Device Identification Register.

global_debug_rom.ROMPIDR**Type**

int

Default value

0x4000bb000

Value of Debug Rom Peripheral Identification Register.

global_debug_rom.ROMPRIDR0**Type**

int

Default value

0x1

Value of Debug ROM Power RequestID Register.

gpccr_el3_gpcp_behaviour**Type**

int

Default value

0x2

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

Number of separate GPT TLB entries.

gpt_walkcache_size**Type**

int

Default value

0x0

Number of GPT walk cache entries.

hardware_translation_table_update_implemented**Type**

int

Default value

0x1

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

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

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

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

Enable 16-bit ASIDs.

has_16bit_vmids**Type**

int

Default value

0x1

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

Implement the 16k LPAE translation granule.

has_4k_granule**Type**

bool

Default value

0x1

Implement the 4k LPAE translation granule.

has_52bit_address_with_16k**Type**

int

Default value

0x0

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

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

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

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

Implement the 64k LPAE translation granule.

has_aarch32_dbgdidr_etc**Type**

bool

Default value

0x1

DBGDIDR, DBGDRAR, DBGDSAR exist even if EL1 doesn't implement AArch32.

has_aarch32_hpd**Type**

bool

Default value

0x0

If true then hierarchical permission disable is supported in AArch32 (FEAT_AA32HPD).

has_aarch64**Type**

bool

Default value

0x1

All implemented exception levels can run in AArch64.

has_actlr2**Type**

bool

Default value

0x0

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR_EL1[63:32].

has_actlr_virtualisation**Type**

bool

Default value

0x0

If true ACTLR_EL12 is implemented and ACTLR_EL1 supports virtualisation.

has_address_breakpoint_linking**Type**

int

Default value

0x0

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

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

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_arm_v8-1**Type**

bool

Default value

0x0

Implement the ARMv8.1 Extension.

has_arm_v8-2**Type**

bool

Default value

0x0

Implement the ARMv8.2 Extension.

has_arm_v8-3**Type**

bool

Default value

0x0

Implement the ARMv8.3 Extension.

has_arm_v8-4**Type**

bool

Default value

0x0

Implement the ARMv8.4 Extension.

has_arm_v8-5**Type**

bool

Default value

0x0

Implement the ARMv8.5 Extension.

has_arm_v8-6**Type**

bool

Default value

0x0

Implement the ARMv8.6 Extension.

has_arm_v8-7**Type**

bool

Default value

0x0

Implement the Armv8.7 Extension.

has_arm_v8-8**Type**

bool

Default value

0x0

Implement the ARMv8.8 Extension.

has_arm_v8-9**Type**

bool

Default value

0x0

Implement the ARMv8.9 Extension.

has_arm_v9-0**Type**

bool

Default value

0x0

Implement the ARMv9.0 Extension.

has_arm_v9-1**Type**

bool

Default value

0x0

Implement the ARMv9.1 Extension.

has_arm_v9-2**Type**

bool

Default value

0x0

Implement the ARMv9.2 Extension.

has_arm_v9-3**Type**

bool

Default value

0x0

Implement the ARMv9.3 Extension.

has_arm_v9-4**Type**

bool

Default value

0x0

Implement the ARMv9.4 Extension.

has_arm_v9-5**Type**

bool

Default value

0x0

Implement the ARMv9.5 Extension.

has_arm_v9-6**Type**

bool

Default value

0x0

Implement the ARMv9.6 Extension. This feature is incomplete and under development.

has_asid2**Type**

int

Default value

0x1

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

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

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

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

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

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

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

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

If true, implements branch record buffer extension (FEAT_BRBE).

has_brbe_v1p1**Type**

bool

Default value

0x0

If true, implements FEAT_BRBEv1p1.

has_ccidx**Type**

bool

Default value

0x0

Implement the ARMv8.3 CCSIDR Extension. Extending the ccsidr number of sets.

has_cfinv_rmif_setf**Type**

int

Default value

0x1

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

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

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

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

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

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

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

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

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

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

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

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

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

If true, a debug ROM will be generated describing the cluster's debug components.

has_delayed_brbe_records**Type**

bool

Default value

0x1

If true, a synchronization barrier is required to update the BRB records (FEAT_BRBE).

has_delayed_ctireg**Type**

bool

Default value

0x0

Delay the functional effect of CTI register writes until ISB or implicit barrier.

has_delayed_dbgreg**Type**

bool

Default value

0x0

Delay the functional effect of external debug register writes until ISB or implicit barrier.

has_delayed_mdscr_el1**Type**

bool

Default value

0x0

Delay the functional effect of MDSCR_EL1 register writes until ISB or implicit barrier.

has_delayed_oslar_el1**Type**

bool

Default value

0x0

Delay the functional effect of OSLAR_EL1 register writes until ISB or implicit barrier.

has_delayed_pmureg**Type**

bool

Default value

0x0

Delay the functional effect of PMU register writes until ISB or implicit barrier.

has_delayed_sysreg**Type**

bool

Default value

0x0

Delay the functional effect of system register writes until ISB or implicit barrier.

has_delayed_wfe_trap**Type**

int

Default value

0x0

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

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

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

Implement ARMv8.5 feature to prevent unprivileged access to one half of the memory (FEAT_EOPD). 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

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

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

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

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

Implement EDACR register.

has_edhsr**Type**

int

Default value

0xf

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

Implements EL2.

has_el3**Type**

bool

Default value

0x1

Implements EL3.

has_enhanced_pac**Type**

bool

Default value

0x0

If pointer authentication is enabled then implement enhanced PAC (FEAT_EPAC).

has_enhanced_pan**Type**

int

Default value

0x1

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

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

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

Whether Enhanced Translation Synchronization is supported (FEAT_ETS).

has_exception_trapping_form_of_vector_catch**Type**

bool

Default value

0x1

Implement the exception trapping form of vector catch debug event.

has_extended_recip_estimate**Type**

int

Default value

0x0

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

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

Implement FEAT_F8F16MM and dependent features.

has_f8f32mm**Type**

bool

Default value

0x0

Implement FEAT_F8F32MM and dependent features.

has_faminmax**Type**

bool

Default value

0x0

Implement FEAT_FAMINMAX.

has_far_not_valid**Type**

bool

Default value

0x0

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

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

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

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

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

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

If true then Fine Grained Write EL3 is enabled (FEAT_FGWTE3).

has_fixed_function_instr_counter**Type**

int

Default value

0x0

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

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

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

Implement FPMR (FEAT_FPMR).

has_fprcvt**Type**

int

Default value

0x1

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

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

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

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

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

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

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

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

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

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

Allow hypervisor to set MDCR_EL2.HPMN to 0 (FEAT_HPMNO) 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

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

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

Implement the optional IT disable feature.

has_ite**Type**

int

Default value

0x0

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

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

Implement the ARMv8 Large System Extensions (FEAT_LSE).

has_large_ttbr_ba_without_lpa**Type**

bool

Default value

0x1

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

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

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

Implement the SCTLR_ELx.LSMAOE (Load/Store Multiple Atomicity and Ordering Enable) and SCTLR_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

If true then it support the RCpc feature from ARMv8.3 (FEAT_LRCPC).

has_lrcpc3

Type

int

Default value

0x0

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

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 ARMv9.5 is enabled. - 2, feature is implemented.

has_lsui

Type

int

Default value

0x1

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

Implement FEAT_LUT.

has_mismatch_and_range_breakpoints**Type**

int

Default value

0x0

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

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

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

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

Implement max-power mitigation mechanism (MPMM).

has_mt_pmu_disable_feature**Type**

int

Default value

0x0

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

Whether MTE asynchronous faults are supported (FEAT_MTE_ASYNC).

has_mte_ctrl_bits_stateful**Type**

bool

Default value

0x0

if memory_tagging_support_level == 1, Whether mte specific control bits in system registers are stateful.

has_mte_perm**Type**

bool

Default value

0x0

Implement tag access permission (FEAT_MTE_PERM).

has_mte_tag_related_fault_high_prio_than_data**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

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

Whether PAR_EL1[10] is RAZ/WI.

has_partial_delayed_mdscr_el1**Type**

bool

Default value

0x0

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

If true, pc sample-based profiling is enabled (FEAT_PCSRv8, FEAT_PCSRv8p2).

has_pc_sample_profiling_enable**Type**

bool

Default value

0x0

Whether PC Sample profiling enable is implemented (FEAT_PCSRv8p9).

has_pcdphint**Type**

int

Default value

0x0

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

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

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

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

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

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

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

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

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

Implement optional PMU extension feature to reserve event counters for external agents from ARMv9.6 (FEAT_PMUv3_EXTPMN). 0 not implemented, 1 implemented Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.6 is enabled. - 2, feature is implemented.

has_pmu_for_sme_extension**Type**

int

Default value

0x0

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

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

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

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

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

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

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

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

Supports QARMA3 pointer authentication algorithm (FEAT_PACQARMA3).

has_ras**Type**

int

Default value

0x0

Implements the ARMv8 RAS Extension. 0 = NO_RAS, 1 = MINIMAL_RAS, 2 = FULL_RAS (FEAT_RAS).

has_ras_aderr**Type**

int

Default value

0x0

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

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

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

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

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

[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

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

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

[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_mmap_view**Type**

bool

Default value

0x0

Implement memory mapped view of RAS Registers for cores.

has_ras_pfar**Type**

int

Default value

0x0

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

[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_restriction_on_speculative_data_loaded**Type**

int

Default value

0x1

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

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

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

If true then RME GPC2 extension is enabled (FEAT_RME_GPC2).

has_rme_gpc3**Type**

bool

Default value

0x0

If true then RME GPC3 extension is enabled (FEAT_RME_GPC3).

has_rndr**Type**

int

Default value

0x0

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

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

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

Support for RPRFM hint instruction (FEAT_RPRFM).

has_sctlr2**Type**

int

Default value

0x1

Implement SCTLR2_ELx registers (FEAT_SCTLR2). 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

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

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

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

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

Implement software lock in memory-mapped CTI, PMU, and external debug interfaces.

has_spe_eft**Type**

int

Default value

0x1

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

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

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

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

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

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

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

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

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

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

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

Implement the extended XN[1:0] stage 2 control from ARMv8.2 for Aarch32 (FEAT_XNX).

has_statistical_profiling**Type**

bool

Default value

0x1

Whether Statistical Based Profiling is implemented (FEAT_SPE).

has_supersections**Type**

bool

Default value

0x1

Whether VMSAv8-32 supersection to support more than 32-bit PA using short descriptor is implemented.

has_sve**Type**

int

Default value

0x0

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

Report asynchronous abort due to unsupported load atomics as synchronous (Cacheable).

has_synchronous_load_atomics_noncacheable**Type**

bool

Default value

0x1

Report asynchronous abort due to unsupported load atomics as synchronous (Non-Cacheable).

has_synchronous_store_atomics**Type**

bool

Default value

0x0

Report asynchronous abort due to unsupported store atomics as synchronous (Cacheable).

has_synchronous_store_atomics_noncacheable**Type**

bool

Default value

0x0

Report asynchronous abort due to unsupported store atomics as synchronous (Non-Cacheable).

has_sysinstr128**Type**

int

Default value

0x0

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

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

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

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

Detected inconsistent TLB content generate aborts.

has_tlb_pa_caching

Type

bool

Default value

0x0

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

Implement support for TLB Range Maintenance instructions (TLBI RVAE1, etc) from ARMv8.4 (FEAT_TLBIRANGE). 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

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

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

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

If true, implements TME, the Transactional Memory Extension (FEAT_TME).

has_translation_hardening**Type**

int

Default value

0x0

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

If true, implements the Trace Buffer Extension (FEAT_TRBE).

has_trbe_ext**Type**

int

Default value

0x0

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

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

Implement support for SCTLRL_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

Report unsupported exclusive access with Unsupported Exclusive fault status (otherwise use external abort).

has_v8_4_debug_extension**Type**

int

Default value

0x1

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

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

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

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

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

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

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

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

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

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

Where FEAT_SPEv1p2 is implemented, whether the inverting filtering by events.

has_v8_7_spe_previous_branch_target**Type**

bool

Default value

0x1

Where FEAT_SPEv1p2 is implemented, whether the optional branch target feature is implemented.

has_v8_8_debug_extension**Type**

int

Default value

0x1

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

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

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

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

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

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

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

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

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

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

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

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

Implement write accesses buffering before L1 cache. May affect ext_abort behaviour.

has_xs

Type

int

Default value

0x1

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

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

If true, HCR_EL2.MIOCNCE is treated as R/W instead of RAZ/WI.

hcr_swio_res1

Type

bool

Default value

0x0

Whether HCR.SWIO and/or HCR_EL2.SWIO are RES1.

hdbss_error_fault_type**Type**

int

Default value

0x0

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

If true, HPFAR_EL2 is set to 0 when IPA is not valid for stage 2 faults.

hsr_uncond_cc**Type**

bool

Default value

0x0

Condition codes reported in HSR as AL if it passes.

hvbar_reset_is_rvbar**Type**

bool

Default value

0x0

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

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

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

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-prefetch_level`

Type

int

Default value

0x0

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

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

L1 I-Cache read bus width in bytes used to calculate per-access timing annotations.

`icache-read_latency`

Type

int

Default value

0x0

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

icache-ways**Type**

int

Default value

0x2

L1 I-Cache number of ways (sets are implicit from size).

id_spec_fpacc_raz**Type**

bool

Default value

0x0

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.

ignore_DBGPRCR_CWRR**Type**

bool

Default value

0x0

Ignore writes to the deprecated DBGPRCR.CWRR bit.

ignore_access_flag_update_by_at_ops**Type**

bool

Default value

0x0

If true, AT operations do not update access flag.

ignore_data_abt_on_af_update_by_at_ops**Type**

bool

Default value

0x1

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

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

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

Whether traps get ignored for the following registers in debug state: * AArch64: MDCCSR_EL0, OSDTREX_EL1, OSDTRTX_EL1, MDCCINT_EL1. * AArch32: DBGDSCRint, DBGDIDR, DBGDSAR, DBGDRAR, DBGDTRRext, DBGDTRTXext, DBGDCCINT.

illegal_state_exception_priority

Type

int

Default value

0x0

IMPDEF priority of Illegal State Exception. 0: After breakpoint exceptions 1: Before Instruction Abort.

imp_def_functionality_behaviour

Type

int

Default value

0x0

Behaviour of IMPLEMENTATION DEFINED registers and system instructions. 0, UNDEF. 1, RAZ/WI.

impdef_regs_and_unpred_from_implementation

Type

string

Default value

""

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

"[]"

Configure mask/reset bitmasks for impdef. registers in a JSON format which is (where 'bitwise' indicates 'reset'/'mask' to flip existing bits): [{ "name": "IMP_SYSREG0_EL1", "reset":0, "mask":0, "encoding":0, "bitwise":true}*].

independent_cache_control_traps**Type**

int

Default value

0x0

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

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

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

internal_vgic**Type**

bool

Default value

0x0

Instantiate VGIC peripheral in this processor.

is_debug_state_pmu_snapshot_allowed**Type**

bool

Default value

0x1

If true, PMU snapshot is allowed in debug state.

is_first_pcsr_sample_ignored**Type**

bool

Default value

0x0

If true, First read of PMPCSR register after reset returns 0xFFFFFFFF.

is_mt_res0**Type**

bool

Default value

0x0

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

If true, SError is edge-triggered. Otherwise, its level-triggered.

is_tagged_nsh_treated_as_tagged**Type**

bool

Default value

0x1

Whether a tagged NonShared memory attribute is treated as tagged or not.

is_uniprocessor**Type**

bool

Default value

0x0

Value for the U bit in MPIDR. true disables L1 cache coherency protocols.

isb_is_branch**Type**

bool

Default value

0x0

If true, ISB is traced as an immediate branch. This allows to count ISB as a branch in debug extensions (e.g. PMU).

ish_is_osh**Type**

bool

Default value

0x0

Whether Innershareable is same as OuterShareable.

itd_conditional_instructions_are_32bit**Type**

bool

Default value

0x0

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

If true, JIDR register access is UNDEF at EL0.

jmcr_is_undef_at_el0**Type**

bool

Default value

0x0

If true, JMCR register access is UNDEF at EL0.

joscr_is_undef_at_el0**Type**

bool

Default value

0x0

If true, JOSCR register access is UNDEF at EL0.

l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

l2cache-read_latency

Type

int

Default value

0x0

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

L2 Cache size in bytes.

l2cache-snoop_data_transfer_latency

Type

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

l2cache-write_access_latency**Type**

int

Default value

0x0

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

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

l2cache-write_latency**Type**

int

Default value

0x0

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

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

L3 Cache MPAMF_AIDR architecture major version.

l3cache-mpamf.arch_minor_ver**Type**

int

Default value

0x0

L3 Cache MPAMF_AIDR architecture minor version.

l3cache-mpamf.bwa_width_ns**Type**

int

Default value

0x10

L3 Cache width of MPAM bandwidth allocation fields for non-secure accesses.

l3cache-mpamf.bwa_width_s**Type**

int

Default value

0x10

L3 Cache width of MPAM bandwidth allocation fields for secure accesses.

l3cache-mpamf.cmax_width_ns**Type**

int

Default value

0x0

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

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

L3 Cache width of MPAM cache portion bitmap for non-secure accesses. If 0, the feature is not implemented, and all ways are available.

l3cache-mpamf.cpbm_width_rl**Type**

int

Default value

0x0

L3 Cache width of MPAM cache portion bitmap for realm accesses. If 0, the feature is not implemented, and all ways are available.

l3cache-mpamf.cpbm_width_rt**Type**

int

Default value

0x0

L3 Cache width of MPAM cache portion bitmap for root accesses. If 0, the feature is not implemented, and all ways are available.

l3cache-mpamf.cpbm_width_s**Type**

int

Default value

0x0

L3 Cache width of MPAM cache portion bitmap for secure accesses. If 0, the feature is not implemented, and all ways are available.

l3cache-mpamf.csu_num_mon_ns**Type**

int

Default value

0x0

L3 Cache number of MPAM cache storage usage monitors for non-secure accesses.

l3cache-mpamf.csu_num_mon_rl**Type**

int

Default value

0x0

L3 Cache number of MPAM cache storage usage monitors for realm accesses.

l3cache-mpamf.csu_num_mon_rt**Type**

int

Default value

0x0

L3 Cache number of MPAM cache storage usage monitors for root accesses.

l3cache-mpamf.csu_num_mon_s**Type**

int

Default value

0x0

L3 Cache number of MPAM cache storage usage monitors for secure accesses.

l3cache-mpamf.esr_mask**Type**

int

Default value

0xffffffff

L3 Cache MPAMF_ESR mask value.

l3cache-mpamf.has_esr**Type**

bool

Default value

0x0

L3 Cache's MPAMF_ESR, MPAMF_ECR, and MPAM error handling implemented.

l3cache-mpamf.has_extd_esr**Type**

bool

Default value

0x0

L3 Cache's MPAMF_ESR is 64-bits.

l3cache-mpamf.has_impl_idr**Type**

bool

Default value

0x0

L3 Cache's MPAMF_IMPL_IDR is present.

l3cache-mpamf.has_mbwu_long_counter**Type**

bool

Default value

0x0

L3 Cache has long MBWU counter and capture registers.

l3cache-mpamf.has_mpamfidr_ext**Type**

bool

Default value

0x0

MPAMF_IDR.EXT support.

l3cache-mpamf.has_partid_nrw**Type**

bool

Default value

0x0

Narrowing part ID register is present. This is global rather than per-instance.

l3cache-mpamf.has_priority_partitioning**Type**

bool

Default value

0x0

The selected resource has priority partitioning described in MPAMF_PRI_IDR.

l3cache-mpamf.has_prod_id**Type**

int

Default value

0x0

L3 Cache MPAMF_IIDR product ID supported.

l3cache-mpamf.has_prod_rev**Type**

int

Default value

0x0

L3 Cache MPAMF_IIDR product REVISION supported.

l3cache-mpamf.has_prod_var**Type**

int

Default value

0x0

L3 Cache MPAMF_IIDR product VARIENT supported.

l3cache-mpamf.has_prop_ns**Type**

bool

Default value

0x0

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

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

L3 Cache has resource instance selection support.

l3cache-mpamf.max_partid_ns**Type**

int

Default value

0xffff

L3 Cache Maximum value of non-secure PARTID supported.

l3cache-mpamf.max_partid_rl**Type**

int

Default value

0xffff

L3 Cache Maximum value of realm PARTID supported for RME implementations.

l3cache-mpamf.max_partid_rt**Type**

int

Default value

0xffff

L3 Cache Maximum value of root PARTID supported for RME implementations.

l3cache-mpamf.max_partid_s**Type**

int

Default value

0xffff

L3 Cache Maximum value of secure PARTID supported.

l3cache-mpamf.max_pmg_ns**Type**

int

Default value

0xff

L3 Cache Maximum value of non-secure PMG supported.

l3cache-mpamf.max_pmg_rl**Type**

int

Default value

0xff

L3 Cache Maximum value of realm PMG supported for RME implementations.

l3cache-mpamf.max_pmg_rt**Type**

int

Default value

0xff

L3 Cache Maximum value of root PMG supported for RME implementations.

l3cache-mpamf.max_pmg_s**Type**

int

Default value

0xff

L3 Cache Maximum value of secure PMG supported.

l3cache-mpamf.mbwu_long_counter_width**Type**

int

Default value

0x0

L3 Cache long MBWU counter width. 0: 63 bits, 1: 44 bits.

l3cache-mpamf.no_impl_msmon**Type**

bool

Default value

0x0

L3 Cache's MPAMF_IMPL_IDR does not describe resource monitors.

l3cache-mpamf.no_impl_part**Type**

bool

Default value

0x0

L3 Cache's MPAMF_IMPL_IDR does not describe resource partitioning controls.

l3cache-mpamf.ris_max**Type**

int

Default value

0x0

L3 Cache's largest resource instance selector value defined.

l3cache-mpamf_base**Type**

int

Default value

0x0

L3 Cache memory mapped MPAM registers base address.

l3cache-read_access_latency**Type**

int

Default value

0x0

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

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

l3cache-read_latency**Type**

int

Default value

0x0

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

`l3cache-write_access_latency`**Type**

int

Default value

0x0

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

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

l3cache-write_latency**Type**

int

Default value

0x0

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

Whether exception catch is traced as part of exception entry/exit in same cycle.

log2_trace_buffer_alignment**Type**

int

Default value

0x0

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

ls64_ignore_s1_unpred_memattr_transformation**Type**

bool

Default value

0x0

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**

bool

Default value

0x0

If true, FEAT_LS64 single-copy atomic 64-byte load/store instructions' memory attributes check is done on the combined memory attributes at the end of all enabled translation stages.

max_32bit_el**Type**

int

Default value

0x3

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

MDRAR_EL1 is RES0.

mdselr_le_16_bps_wps_behaviour**Type**

int

Default value

0x0

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.

mec_support_level**Type**

int

Default value

0x0

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

Used with `memory.acp.AxCACHE_pattern` to define which memory types the ACP port accepts. All transactions which do not satisfy $(\text{AxCACHE} \& \text{mask}) == \text{pattern}$ will abort.

`memory.acp.AxCACHE_pattern`

Type

int

Default value

0x0

Used with `memory.acp.AxCACHE_mask` to define which memory types the ACP port accepts. All transactions which do not satisfy $(\text{AxCACHE} \& \text{mask}) == \text{pattern}$ will abort.

`memory.l2_cache.is_inner_cacheable`

Type

bool

Default value

0x1

L2 cache obeys inner cacheable attributes (rather than outer cacheable attributes).

`memory.l2_cache.is_inner_shareable`

Type

bool

Default value

0x1

L2 cache obeys inner shareable attributes (rather than outer shareable attributes).

`memory_tagging_support_level`

Type

int

Default value

0x0

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

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

Block size used for memcpy memory accesses.

mops_cpy_default_dir**Type**

int

Default value

0x0

Default direction for non-overlapping memcpy operations: 0, forwards. 1, backwards.

mops_cpy_handle_async_exceptions**Type**

bool

Default value

0x0

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

Percentage of data copied in memcpy 'E' instructions.

mops_cpy_pre_size**Type**

int

Default value

0xa

Percentage of data copied in memcpy 'P' instructions.

mops_cpy_pre_size_threshold**Type**

int

Default value

0x0

Size threshold in Bytes for CPY*P* instructions.

mops_cpy_single_access**Type**

bool

Default value

0x0

Execute memcpy as a single read and single write access.

mops_cpy_write_abort_before_read**Type**

bool

Default value

0x0

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

Fault because of mismatched implementation option when the operation is of size 0.

mops_exec_order_can_fault**Type**

bool

Default value

0x0

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

Fault because of mismatched implementation option when inst_cpy_size is 0.

mops_inst_set_zero_size_can_fault**Type**

bool

Default value

0x1

Fault because of mismatched implementation option when inst_set_size is 0.

mops_mismatched_page_crossing_access_unpred**Type**

int

Default value

0x0

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

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

Block size used for memset memory accesses.

mops_set_handle_async_exceptions**Type**

bool

Default value

0x0

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

Percentage of data copied in memset 'E' instructions.

mops_set_pre_size**Type**

int

Default value

0xa

Percentage of data copied in memset 'P' instructions.

mops_set_single_access**Type**

bool

Default value

0x0

Execute memset as a single read and single write access.

mops_set_zero_size_can_fault**Type**

bool

Default value

0x1

Fault because of mismatched implementation option when the operation is of size 0.

mops_setg_unaligned_does_mismatch_fault**Type**

bool

Default value

0x0

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

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

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

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

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

MPAM MPAMBWIDR_EL1.US_FRAC: The fractional part of the window width in microseconds.

mpam_bw_us_int**Type**

int

Default value

0x0

MPAM MPAMBWIDR_EL1.US_INT: The integer part of the window width in microseconds.

mpam_force_ns_rao**Type**

bool

Default value

0x0

Whether MPAM3_EL3.FORCE_NS bit is RAO/WI.

mpam_frac**Type**

int

Default value

0x0

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_MPAMv0p1 has_mpam = true, mpam_frac = 0 -> FEAT_MPAMv1p0 has_mpam = true, mpam_frac = 1 -> FEAT_MPAMv1p1.

mpam_has_altsp**Type**

bool

Default value

0x0

MPAM Whether MPAMIDR_EL1.HAS_ALTSP bit is set or clear.

mpam_has_bw_ctrl**Type**

bool

Default value

0x0

MPAM Whether MPAMIDR_EL1.HAS_BW_CTRL bit is set or clear.

mpam_has_hcr**Type**

bool

Default value

0x0

MPAM Whether MPAMIDR_EL1 HAS_HCR bit is set or clear.

mpam_max_partid**Type**

int

Default value

0xffff

MPAM Maximum PARTID Supported.

mpam_max_pmg**Type**

int

Default value

0xff

MPAM Maximum PMG Supported.

mpam_max_vpmr**Type**

int

Default value

0x0

MPAM Maximum VPMR Supported.

mpamidr_has_force_ns**Type**

int

Default value

0x0

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

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

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

Layout of MPIDR. 0 AFF0 is CPUID, 1 AFF1 is CPUID.

mpmm_accumulator_multiplier

Type

int

Default value

0x1

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

""

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_G0_TP":2, "MPMM_G0_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"

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

Value of CTR_EL0.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

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

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

Number of Limited Ordering Region descriptors implemented (if ARM v8.1 extensions are implemented) (FEAT_LOR).

num_loregions**Type**

int

Default value

0x0

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

if FEAT_ABLE is implemented, Number of address matching breakpoints that support address linking.

number_of_error_records**Type**

int

Default value

0x0

Cores Number of Error records supported for RAS.

nv_frac_support_level**Type**

int

Default value

0x0

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

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"`

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

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

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).

par_ns_set_unknown_bit**Type**

bool

Default value

0x1

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.

pfar_is_valid**Type**

bool

Default value

0x1

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

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

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

Defines whether the address translation performed by the Profiling Buffer manages the Access Flag and dirty state.

pmcr_disable_events_export**Type**

bool

Default value

0x1

If true, export for PMU events is disabled. This configures PMCFGR.EX field.

pmeocr_res0_fields_razwi**Type**

bool

Default value

0x0

If true, RES0 fields of PMEOCR_EL1 are accessed with a Read-As-Zero/Write-Ignore policy.

pmmir_el1_bus_slots**Type**

int

Default value

0x0

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

Width, in bytes, of accesses counted by BUS_ACCESS. From v8.7 PMU extension.

pms_idr_max_size**Type**

int

Default value

0x6

Defines largest size for a single SPE record (rounded up to a power of 2).

pmu-num_counters**Type**

int

Default value

0x8

Number of PMU counters implemented.

pmu_async_exception_delay**Type**

int

Default value

0x0

Configure PMU asynchronous exception delay in CPU cycles (FEAT_SEBEP).

pmu_cycle_counter_counts_actual_cycles**Type**

bool

Default value

0x0

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

PMU (if present) implements event number 0x1e, CHAIN.

pmu_precise_events**Type**

string

Default value

""

"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

Implement FEAT_PMUv3_TH and if so the width of PMEVTYPER<n>_ELO.TH in bits. 0, not implemented. 1-12 number of bits in PMEVTYPER<n>_ELO.TH.

poison_range_end_addr

Type

int

Default value

0x0

End PA of poisoned range.

poison_range_start_addr

Type

int

Default value

0x0

Start PA of poisoned range.

pseudo_fault_generation_feature_register

Type

string

Default value

""

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.

pstate_pm_reset

Type

bool

Default value

0x0

Reset value of PSTATE.PM.

pstate_ssbs_reset**Type**

bool

Default value

0x0

Reset value of pstate.ssbs.

pstate_ssbs_type**Type**

int

Default value

0x0

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

randomize_unknowns_at_reset**Type**

bool

Default value

0x0

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

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_err_registers_undef_if_no_error_records**Type**

bool

Default value

0x0

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

If true, ERRSELR_EL1 will be undefined if ERRIDR_EL1 indicates that zero error records are implemented.

ras_extra_configurations**Type**

string

Default value

""

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

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

0x2

Log2 of the RAS fault granule size in KB.

ras_mmap_address**Type**

string

Default value

""

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

RAS Pseudo-Fault generation clock rate in MHz.

ras_report_aligned_pa_in_pfar**Type**

bool

Default value

0x0

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

Data used to fill register bits when they become UNKNOWN at reset.

register_reset_data_hi**Type**

int

Default value

0x0

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

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

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

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

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

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

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

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

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

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

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

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

Whether RGSr read of RES0 bits return last written value.

rme_default_mecid_nonsecure**Type**

int

Default value

0x0

Default MECID value for NON-SECURE PAS.

rme_default_mecid_realm**Type**

int

Default value

0x0

Default MECID value for REALM PAS.

rme_default_mecid_root**Type**

int

Default value

0x0

Default MECID value for ROOT PAS.

rme_default_mecid_secure**Type**

int

Default value

0x0

Default MECID value for SECURE PAS.

rme_full_is_tagged_nsh_treated_as_tagged**Type**

bool

Default value

0x0

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

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

Width of MECID in bits.

rme_nsh_cacheable_is_shareable**Type**

bool

Default value

0x0

If true, NSH cacheable becomes shareable cacheable (FEAT_RME).

rme_support_level**Type**

int

Default value

0x0

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).

rmr_always_implemented**Type**

bool

Default value

0x0

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

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

If true, s1 alignment fault has priority over s2 permission faults.

scheduler_mode**Type**

int

Default value

0x0

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

Whether SCR.nET is writeable. Writing to it is purely cosmetic (nET behavior not implemented).

scramble_unknowns_at_reset**Type**

bool

Default value

0x1

Will fill in unknown bits in registers at reset with register_reset_data.

seerror_clear_delay**Type**

int

Default value

0x0

Delay for clearing of SError if SError is level-triggered, in cpu cycles.

set_mops_option**Type**

int

Default value

0x0

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

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

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

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

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

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

Whether SPSR_EL3 is mapped to AArch32 register SPSR_mon.

spsr_m4_res0**Type**

bool

Default value

0x0

Whether SPSR_ELx.M[4] bit should be RES0 for AARCH64 only implementations.

stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

stage1_tlb_size**Type**

int

Default value

0x0

Number of stage1 only tlb entries.

stage1_walkcache_size**Type**

int

Default value

0x0

Number of stage1 only walk cache entries.

stage2_tlb_size**Type**

int

Default value

0x0

Number of stage2 only tlb entries.

stage2_walkcache_size**Type**

int

Default value

0x0

Number of stage2 only walk cache entries.

statistical_profiling_buffer_alignment**Type**

int

Default value

0x1

Statistical profiling alignment constraint for sample buffer.

statistical_profiling_random_interval_is_separate**Type**

bool

Default value

0x0

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

Statistical profiling recommended minimum sampling interval.

stex_fail_suppress_sync_data_aborts**Type**

bool

Default value

0x0

If true, synchronous data aborts are not reported if store exclusive fails.

store_excl_fail_tag_check_action**Type**

int

Default value

0x0

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

If true, a strex fail can hit watchpoint.

stzgm_reports_fault_address_from_reg_arg**Type**

int

Default value

0x0

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

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

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

Attempt to combine the execution of MOVPRFX and the destructively-encoded instruction that follows it.

sve.disable_speculative_accesses

Type

bool

Default value

0x0

All speculative memory accesses behave as though faulting, without accessing memory.

sve.enable_at_reset

Type

bool

Default value

0x0

Start with system registers set up for Scalable Vector Extension use.

sve.ffr_16b_pattern_UNKNOWN

Type

int

Default value

0x0

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

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

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

Set ESR_ELx.TFV during FP exception. Trapped exception flags are valid.

sve.fp_exception_set_vecitr**Type**

bool

Default value

0x0

If true, set ESR_ELx.VECITR during FP exception. Otherwise, set RES0.

sve.has_b16b16**Type**

int

Default value

0x0

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

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

Whether SME is implemented (FEAT_SME).

sve.has_sme2**Type**

bool

Default value

0x0

Whether SME2 is implemented (FEAT_SME2).

sve.has_sme_b16b16**Type**

int

Default value

0x0

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

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

If SME is implemented, whether double-precision FMOPA and FMOPS are implemented.

sve.has_sme_f8f16**Type**

int

Default value

0x1

If SME2 is implemented, whether FEAT_SME_F8F16 is implemented.

sve.has_sme_f8f32**Type**

int

Default value

0x1

If SME2 is implemented, whether FEAT_SME_F8F32 is implemented.

sve.has_sme_fa64**Type**

bool

Default value

0x0

Whether FEAT_SME_FA64 is implemented.

sve.has_sme_i16i64**Type**

int

Default value

0x1

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

Whether FEAT_SME_LUTv2 is implemented.

sve.has_sme_priority_control**Type**

bool

Default value

0x1

Whether SME Priority Control is implemented.

sve.has_ssve_aes**Type**

int

Default value

0x0

Indicates support for SVE2 and SME2 AES instructions when the PE is in Streaming SVE mode (FEAT_SSVE_AES).

sve.has_ssve_f8f16mm**Type**

int

Default value

0x0

Implement the FP8 to half-precision matrix multiply instruction when the PE is in Streaming SVE mode (FEAT_SSVE_F8F16MM).

sve.has_ssve_f8f32mm**Type**

int

Default value

0x0

Implement the SVE FP8 to single-precision matrix multiply instruction when the PE is in Streaming SVE mode (FEAT_SSVE_F8F32MM).

sve.has_sve2**Type**

bool

Default value

0x0

Whether SVE2 is implemented (FEAT_SVE2).

sve.has_sve2_aes**Type**

int

Default value

0x2

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

If SVE2 is implemented, whether BitPerm instructions are implemented (FEAT_SVE_BitPerm).

sve.has_sve2_sha3**Type**

bool

Default value

0x1

If SVE2 is implemented, whether SHA3 instructions are implemented (FEAT_SVE_SHA3).

sve.has_sve2_sm4**Type**

bool

Default value

0x1

If SVE2 is implemented, whether SM4 instructions are implemented (FEAT_SVE_SM4).

sve.has_sve_bf16**Type**

bool

Default value

0x1

Whether SVE BFloat16 instructions are implemented.

sve.has_sve_extended_bf16**Type**

int

Default value

0x2

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

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

Whether the SVE FP32 Matrix Multiply instructions are implemented (FEAT_F32MM).

sve.has_sve_mm_f64**Type**

bool

Default value

0x1

Whether the SVE FP64 Matrix Multiply instructions are implemented (FEAT_F64MM).

sve.has_sve_mm_i8**Type**

bool

Default value

0x1

Whether the SVE Int8 Matrix Multiply instructions are implemented (FEAT_I8MM).

sve.movprfx_unpredictable_behavior**Type**

int

Default value

0x0

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

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

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

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

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

The version of SME2 if implemented. Possible values are: 0 - FEAT_SME2, 1 - FEAT_SME2p1, 2 - SME2_2024 (Note: This will change post Alpha).

sve.sme_highest_implemented_priority**Type**

int

Default value

0x0

When SME Priority Control and SME2p2 are implemented, controls the highest implemented priority.

sve.sme_only**Type**

bool

Default value

0x0

If SME is implemented, whether SVE functionality is available only when SM=1.

sve.sme_ssve_fp8_support_level**Type**

int

Default value

0x0

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

Which SME vector lengths are implemented. Represented as a bitfield where $\text{bit}[n]=1$ implies SME vector length of $128 \cdot 2^n$ bits is implemented.

sve.smidr_el1_implementer_val**Type**

int

Default value

0x41

The value of SMIDR_EL1.Implementer.

sve.smidr_el1_revision_val**Type**

int

Default value

0x0

The value of SMIDR_EL1.Revision.

sve.sve2_version**Type**

int

Default value

0x0

The version of SVE2 if implemented. Possible values are: 0 - FEAT_SVE2, 1 - FEAT_SVE2p1, 2 - SVE2_2024 (Note: This will change post Alpha).

sve.sve_dabt_far_behaviour**Type**

int

Default value

0x0

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

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.

sve.trace_za_tilewise

Type

bool

Default value

0x1

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

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

Simulated value for a state that has an UNKNOWN value after reset.

sve.vecLEN

Type

int

Default value

0x8

SVE vector length in units of 64 bits.

sve.z_reg_on_load_fault_behaviour

Type

int

Default value

0x0

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

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

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

If true, swp with dest as xzr is treated as store atomic.

sync_ext_abort_is_sync_serror**Type**

bool

Default value

0x0

Treat synchronous external aborts as synchronous SErrors (RASv8.9). 0, synchronous external abort. 1, synchronous serror.

system_pmu_id**Type**

int

Default value

0x0

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

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

UNDEF exception is taken even if condition code check fails.

tcr_ps_reserved_value_size**Type**

int

Default value

0x0

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

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

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

TIDCP has priority over UNDEF for accesses to IMPLEMENTATION DEFINED functionality from EL0.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_or_ic_invalid_xt**Type**

int

Default value

0x0

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

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

If true, disables the read/write set related optimizations of the model.

tme_imp-failures-can-retry

Type

bool

Default value

0x0

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

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

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

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

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

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

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

If true, executing WFE inside a transaction will cause it to fail with IMP=1.

tme_write-set-size**Type**

int

Default value

0x0

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

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

ETM trace registers support access via system registers.

trace_icc_registers_as_icv_when_redirected**Type**

bool

Default value

0x0

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

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).

trap_dc_cmo_to_pou_if_nop**Type**

bool

Default value

0x1

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

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

Whether setting HCR_EL2.TID3 traps reserved group3 id registers.

trbe_cmod**Type**

int

Default value

0x0

TRBE Customer Modified.

trbe_des**Type**

int

Default value

0x0

Designer, JEP106 identification code.

trbe_external_abort_handling**Type**

int

Default value

0x0

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

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

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

TRBE MPAM support.

trbe_part**Type**

int

Default value

0x0

Part number.

trbe_partid_max**Type**

int

Default value

0x0

Largest permitted TRBDEVID1.PARTID value.

trbe_pmg_max**Type**

int

Default value

0x0

Largest permitted TRBDEVID1.PMG value.

trbe_revand**Type**

int

Default value

0x0

TRBE component minor revision.

trbe_revision**Type**

int

Default value

0x0

TRBE architecture revision.

treat-dcache-cmos-to-occ-as-nop**Type**

bool

Default value

0x0

Implement CMOs to Outer cache level as NOP.

treat-dcache-cmos-to-poc-as-nop**Type**

int

Default value

0x0

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

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

Treat data cache invalidate operations as clean and invalidate.

treat-icache-cmos-to-pou-as-nop**Type**

int

Default value

0x0

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

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

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

If true, treat PLD as NOP.

`treat_pli_as_nop`

Type

bool

Default value

0x0

If true, treat PLI as NOP.

`treat_wfi_wfe_as_nop`

Type

bool

Default value

0x0

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

On Illegal ERET to AArch32, truncate PC to 32-bits.

undef_ccsidr2_access_for_unimplemented_aarch32

Type

bool

Default value

0x0

Whether access to CCSIDR2 is undef if AArch32 is implemented or not at EL1.

unification-level

Type

int

Default value

0x1

Level of Unification Inner Shareable for the cache hierarchy.

unification-uniprocessor-level

Type

int

Default value

0x1

Level of Unification Uniprocessor for the cache hierarchy.

unpred_LSE128_overlap

Type

int

Default value

0x1

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

If true, BRB IALL/INJ instruction will behave as UNDEFINED if Xt != 0b11111.

unpred_clear_ISV_for_exception_before_software_step**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

HVC unpredictable behaviour. 0, UNDEF. 1, NOP.

unpredictable_smc_behaviour

Type

int

Default value

0x0

SMC unpredictable behaviour. 0, UNDEF. 1, NOP.

unpredictable_wfet_and_wfit_behaviour

Type

int

Default value

0x1

WFET and WFIT unpredictable behaviour in debug state. 0, UNDEFINED. 1, NOP.

unsupported_atomic_fault_type

Type

int

Default value

0x0

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

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_Xt_as_LDG_STG_input**Type**

bool

Default value

0x1

Use new MTE Instructions formats for LDG/ST(Z)(2)G, which use Xt as an input.

use_architectural_names**Type**

bool

Default value

0x0

Use names SP/LR/PC instead of R13/R14/R15.

use_mte_eac_02_instructions_encoding**Type**

bool

Default value

0x1

Use new MTE Instructions encoding since MTE spec EAC 0.2 in ARMv8.5.

use_mte_eac_08_tfsr_encoding**Type**

bool

Default value

0x1

Use new MTE TFSR_ELx encodings since MTE spec EAC 0.8 in ARMv8.5.

use_rosetta_disass**Type**

int

Default value

0x1

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

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

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

Translation table entries with the contiguous hint bit set generate large TLB entries.

user_defined_rom_table_debug_power_config**Type**

string

Default value

""

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

VPU data path width.

walk_cache_latency**Type**

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

warn_unpredictable_in_v7**Type**

bool

Default value

0x1

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

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

Configure WFE wakeup delay in CPU cycles.

wfi_wakeup_delay**Type**

int

Default value

0x0

Configure WFI wakeup delay in CPU cycles.

wnr_is_read_for_s2f_on_s1_atomic_instr_fault**Type**

bool

Default value

0x0

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

Whether WnR is 0 for stage2 fault on stage1 descriptor dbm update.

wp_ignores_dbm_update**Type**

bool

Default value

0x0

If true, dbm update is ignored on watchpoint hit.

wp_num_reporting**Type**

int

Default value

0x0

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 3-196: 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 23.

Iris and MTI instances for ARMAEMv8MCT

This model has the following Iris instances:

Table 3-197: 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 3-198: 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

ARMAEMv8MCT contains the following CADI targets:

- ARM_AEMv8M

Ports for ARMAEMv8MCT

Table 3-199: 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 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
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

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

If true, set the bit after reset (as if by IMP_DEF mechanism). Ignored if SECEXT=false.

AIRCR.BFHFNMINWritable**Type**

bool

Default value

0x1

Is AIRCR.BFHFNMIN bit[13] writeable.

AIRCR.ENDIANNESS**Type**

bool

Default value

0x0

Initialize processor to big endian mode.

AIRCR.PRISWritable**Type**

bool

Default value

0x1

Is AIRCR.PRIS bit[14] writeable.

AIRCR.VECTCLRACTIVEChangesMode**Type**

bool

Default value

0x1

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

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

Is AIRCR_NS.DIT bit[4] writeable.

AIRCR_S.DIT_reset**Type**

bool

Default value

0x1

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

Is AIRCR_S.DIT bit[4] writeable.

BB_PRESENT**Type**

bool

Default value

0x0

Enable bitbanding.

BEATS_PER_TICK**Type**

int

Default value

0x2

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

BF instruction executes as NOP, even if we have LO_BRANCH_INFO.

CCR.BP**Type**

bool

Default value

0x1

Reset value of the Configuration and Control Register's branch prediction enable bit.

CCR.BP_writable**Type**

bool

Default value

0x0

Whether it is possible to modify the Configuration and Control Register's branch prediction enable bit.

CDEMAPPEDONCP**Type**

int

Default value

0xff

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

CDERTLID**Type**

int

Default value

0x20

Value of ID_AFR0.CDERTLID.

CFGMEMALIAS**Type**

int

Default value

0x0

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

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

CFGPAHBSZ**Type**

int

Default value

0x0

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

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

CPSPRESENT**Type**

int

Default value

0xff

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

CPUID**Type**

int

Default value

0x0

Set SCS CPUID Base Register. If set to zero, a default CPUID is used.

CTI**Type**

bool

Default value

0x0

CTI (Cross Trigger Interface) included.

CTI_irq0_pin**Type**

int

Default value

0x4

CTI interrupt request 0 pin.

CTI_irq1_pin**Type**

int

Default value

0x5

CTI interrupt request 1 pin.

DTGU**Type**

bool

Default value

0x0

DTCM Security Gate Unit included.

DTGUBLKSZ**Type**

int

Default value

0x3

DTCM gate unit block size. Size=pow(2, DTGUBLKSZ + 5) bytes.

DTGUMAXBLKS**Type**

int

Default value

0x0

Maximum number of DTCM gate unit blocks. Number of blocks=pow(2, DTGUMAXBLKS).

DWT_CTRL.NOCYCCNT**Type**

bool

Default value

0x0

DWT cycle-counter not present (v8M_bl/v6M never have one).

DWT_CTRL.NOPRFCNT**Type**

bool

Default value

0x0

DWT performance-counters not present (v8M_bl/v6M never have them).

DWT_CTRL.NUMCOMP**Type**

int

Default value

0x4

Number of watchpoint unit comparators implemented.

DWT_DEVARCH.REVISION**Type**

int

Default value

0x1

0: V2, 1: V2.1.

DWT_FUNCTION0.ID**Type**

int

Default value

0xb

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

DWT_VMASK register reset value.

ERRDEVID.NUM**Type**

int

Default value

0x38

RAS: Number of implemented error record indexes, 0 to 56.

FPB_HAS_LSR**Type**

bool

Default value

0x1

FPB has LAR and LSR for software lock if mainline.

FP_CTRL.NUM_CODE**Type**

int

Default value

0x8

Number of breakpoint unit comparators implemented (limited to 15 in V6M or baseline).

FP_CTRL.NUM_LIT**Type**

int

Default value

0x0

How many Literals FPB supports remapping (ignored if baseline or TZM).

FP_REMAP.RMPSPT**Type**

bool

Default value

0x1

FPB supports remapping (ignored if baseline or SECEXT).

ID_DFR0.Debug_Model_M_profile**Type**

bool

Default value

0x1

Set whether debug extensions are implemented.

ID_ISAR0.CmpBranch**Type**

int

Default value

0x3

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

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

level of support for extend instructions.

ID_ISAR1.interwork_instrs**Type**

int

Default value

0x2

level of support for Interworking instructions.

ID_ISAR2.MultiAccessInt**Type**

int

Default value

0x2

level of support for interruptible multi-access instructions.

ID_ISAR2.multS_instrs**Type**

int

Default value

0x3

level of support for advanced signed Multiply instructions.

ID_ISAR2.multU_instrs**Type**

int

Default value

0x2

level of support for advanced unsigned Multiply instructions.

ID_ISAR3.SIMD_instrs**Type**

int

Default value

0x3

level of support for SIMD instructions.

ID_ISAR3.saturate_instrs**Type**

int

Default value

0x1

level of support for saturate instructions.

ID_ISAR3.synchprim_instrs**Type**

int

Default value

0x1

level of support for synchronization primitives ID_ISAR3.

ID_ISAR4.synchPrim_instrs_frac**Type**

int

Default value

0x3

level of support for synchronization primitives ID_ISAR4.

ID_ISAR4.unpriv_instrs**Type**

int

Default value

0x2

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

level of support for instructions with shifts.

ID_ISAR5.PACBTI**Type**

int

Default value

0x0

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

Auxiliary registers bits in ID_MMFR0, indicate the support for Auxiliary registers.

ID_MMFR0.Outermost_shareability**Type**

int

Default value

0x0

Outermost shareability bits in ID_MMFR0, indicate the outermost shareability domain implemented.

ID_MMFR0.ShareLvl**Type**

int

Default value

0x1

Shareability levels bits in ID_MMFR0, indicate the number of Shareability levels implemented.

INITPAHBEN**Type**

bool

Default value

0x0

The P-AHB enable state at reset.

INITVTOR_NS**Type**

int

Default value

0x0

Non-Secure vector-table offset at reset.

INITVTOR_S**Type**

int

Default value

0x0

Secure vector-table offset at reset.

IOP**Type**

bool

Default value

0x0

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

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

IRQDIS1**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

IRQDIS10**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

IRQDIS11**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

IRQDIS12**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

IRQDIS13**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

IRQDIS14**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

IRQDIS15**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+480].

IRQDIS2**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

IRQDIS3**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

IRQDIS4**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

IRQDIS5**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

IRQDIS6**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

IRQDIS7**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

IRQDIS8**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

IRQDIS9**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

ITGU**Type**

bool

Default value

0x0

ITCM Security Gate Unit included.

ITGUBLKSZ**Type**

int

Default value

0x3

ITCM gate unit block size. Size=pow(2, ITGUBLKSZ + 5) bytes.

ITGUMAXBLKS**Type**

int

Default value

0x0

Maximum number of ITCM gate unit blocks. Number of blocks=pow(2, ITGUMAXBLKS).

ITM**Type**

bool

Default value

0x1

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included (unless baseline).

ITM_HAS_LSR**Type**

bool

Default value

0x1

ITM support LAR and LSR for software lock.

LOCKDTGU**Type**

bool

Default value

0x0

Lock down of Data TGU registers write.

LOCKITGU**Type**

bool

Default value

0x0

Lock down of Instruction TGU registers write.

LOCKTCM**Type**

bool

Default value

0x0

Lock down of TCM registers write.

LOCK_NS_MPU**Type**

bool

Default value

0x0

Lock down of Non-Secure MPU registers write.

LOCK_SAU**Type**

bool

Default value

0x0

Lock down of SAU registers write.

LOCK_S_MPU**Type**

bool

Default value

0x0

Lock down of Secure MPU registers write.

LVL_WIDTH**Type**

int

Default value

0x3

Number of bits of interrupt priority (baseline has 2).

MEMORY_REGION_MASK**Type**

int

Default value

0xffffffff

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

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

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

MVE**Type**

int

Default value

0x2

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

Support 8-byte floats.

MVFR1.FP16**Type**

bool

Default value

0x1

FP extension implements half-precision floating-point operations. 0 = Not supported; 1 = Supported.

MVFR1.FPHP**Type**

int

Default value

0x2

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

DEPRECATED: Use parameter MVE instead.

NUM_IRQ**Type**

int

Default value

0x10

Number of user interrupts.

NVIC_ITNS0**Type**

string

Default value

""

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

""

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

""

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

""

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

""

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

""

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

""

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

""

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

""

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

""

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

""

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

""

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

""

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

""

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

""

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

""

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"`

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"

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

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

Enable SAU at reset.

SAU_REGION0.BADDR**Type**

int

Default value

0x0

Base address of SAU region0 at reset.

SAU_REGION0.ENABLE**Type**

bool

Default value

0x0

Enable SAU region0 at reset.

SAU_REGION0.LADDR**Type**

int

Default value

0x0

Limit address of SAU region0 at reset.

SAU_REGION0.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region0 at reset.

SAU_REGION1.BADDR**Type**

int

Default value

0x0

Base address of SAU region1 at reset.

SAU_REGION1.ENABLE**Type**

bool

Default value

0x0

Enable SAU region1 at reset.

SAU_REGION1.LADDR**Type**

int

Default value

0x0

Limit address of SAU region1 at reset.

SAU_REGION1.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region1 at reset.

SAU_REGION10.BADDR**Type**

int

Default value

0x0

Base address of SAU region10 at reset.

SAU_REGION10.ENABLE**Type**

bool

Default value

0x0

Enable SAU region10 at reset.

SAU_REGION10.LADDR**Type**

int

Default value

0x0

Limit address of SAU region10 at reset.

SAU_REGION10.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region10 at reset.

SAU_REGION100.BADDR**Type**

int

Default value

0x0

Base address of SAU region100 at reset.

SAU_REGION100.ENABLE**Type**

bool

Default value

0x0

Enable SAU region100 at reset.

SAU_REGION100.LADDR**Type**

int

Default value

0x0

Limit address of SAU region100 at reset.

SAU_REGION100.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region100 at reset.

SAU_REGION101.BADDR**Type**

int

Default value

0x0

Base address of SAU region101 at reset.

SAU_REGION101.ENABLE**Type**

bool

Default value

0x0

Enable SAU region101 at reset.

SAU_REGION101.LADDR**Type**

int

Default value

0x0

Limit address of SAU region101 at reset.

SAU_REGION101.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region101 at reset.

SAU_REGION102.BADDR**Type**

int

Default value

0x0

Base address of SAU region102 at reset.

SAU_REGION102.ENABLE**Type**

bool

Default value

0x0

Enable SAU region102 at reset.

SAU_REGION102.LADDR**Type**

int

Default value

0x0

Limit address of SAU region102 at reset.

SAU_REGION102.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region102 at reset.

SAU_REGION103.BADDR**Type**

int

Default value

0x0

Base address of SAU region103 at reset.

SAU_REGION103.ENABLE**Type**

bool

Default value

0x0

Enable SAU region103 at reset.

SAU_REGION103.LADDR**Type**

int

Default value

0x0

Limit address of SAU region103 at reset.

SAU_REGION103.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region103 at reset.

SAU_REGION104.BADDR**Type**

int

Default value

0x0

Base address of SAU region104 at reset.

SAU_REGION104.ENABLE**Type**

bool

Default value

0x0

Enable SAU region104 at reset.

SAU_REGION104.LADDR**Type**

int

Default value

0x0

Limit address of SAU region104 at reset.

SAU_REGION104.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region104 at reset.

SAU_REGION105.BADDR**Type**

int

Default value

0x0

Base address of SAU region105 at reset.

SAU_REGION105.ENABLE**Type**

bool

Default value

0x0

Enable SAU region105 at reset.

SAU_REGION105.LADDR**Type**

int

Default value

0x0

Limit address of SAU region105 at reset.

SAU_REGION105.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region105 at reset.

SAU_REGION106.BADDR**Type**

int

Default value

0x0

Base address of SAU region106 at reset.

SAU_REGION106.ENABLE**Type**

bool

Default value

0x0

Enable SAU region106 at reset.

SAU_REGION106.LADDR**Type**

int

Default value

0x0

Limit address of SAU region106 at reset.

SAU_REGION106.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region106 at reset.

SAU_REGION107.BADDR**Type**

int

Default value

0x0

Base address of SAU region107 at reset.

SAU_REGION107.ENABLE**Type**

bool

Default value

0x0

Enable SAU region107 at reset.

SAU_REGION107.LADDR**Type**

int

Default value

0x0

Limit address of SAU region107 at reset.

SAU_REGION107.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region107 at reset.

SAU_REGION108.BADDR**Type**

int

Default value

0x0

Base address of SAU region108 at reset.

SAU_REGION108.ENABLE**Type**

bool

Default value

0x0

Enable SAU region108 at reset.

SAU_REGION108.LADDR**Type**

int

Default value

0x0

Limit address of SAU region108 at reset.

SAU_REGION108.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region108 at reset.

SAU_REGION109.BADDR**Type**

int

Default value

0x0

Base address of SAU region109 at reset.

SAU_REGION109.ENABLE**Type**

bool

Default value

0x0

Enable SAU region109 at reset.

SAU_REGION109.LADDR**Type**

int

Default value

0x0

Limit address of SAU region109 at reset.

SAU_REGION109.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region109 at reset.

SAU_REGION11.BADDR**Type**

int

Default value

0x0

Base address of SAU region11 at reset.

SAU_REGION11.ENABLE**Type**

bool

Default value

0x0

Enable SAU region11 at reset.

SAU_REGION11.LADDR**Type**

int

Default value

0x0

Limit address of SAU region11 at reset.

SAU_REGION11.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region11 at reset.

SAU_REGION110.BADDR**Type**

int

Default value

0x0

Base address of SAU region110 at reset.

SAU_REGION110.ENABLE**Type**

bool

Default value

0x0

Enable SAU region110 at reset.

SAU_REGION110.LADDR**Type**

int

Default value

0x0

Limit address of SAU region110 at reset.

SAU_REGION110.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region110 at reset.

SAU_REGION111.BADDR**Type**

int

Default value

0x0

Base address of SAU region111 at reset.

SAU_REGION111.ENABLE**Type**

bool

Default value

0x0

Enable SAU region111 at reset.

SAU_REGION111.LADDR**Type**

int

Default value

0x0

Limit address of SAU region111 at reset.

SAU_REGION111.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region111 at reset.

SAU_REGION112.BADDR**Type**

int

Default value

0x0

Base address of SAU region112 at reset.

SAU_REGION112.ENABLE**Type**

bool

Default value

0x0

Enable SAU region112 at reset.

SAU_REGION112.LADDR**Type**

int

Default value

0x0

Limit address of SAU region112 at reset.

SAU_REGION112.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region112 at reset.

SAU_REGION113.BADDR**Type**

int

Default value

0x0

Base address of SAU region113 at reset.

SAU_REGION113.ENABLE**Type**

bool

Default value

0x0

Enable SAU region113 at reset.

SAU_REGION113.LADDR**Type**

int

Default value

0x0

Limit address of SAU region113 at reset.

SAU_REGION113.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region113 at reset.

SAU_REGION114.BADDR**Type**

int

Default value

0x0

Base address of SAU region114 at reset.

SAU_REGION114.ENABLE**Type**

bool

Default value

0x0

Enable SAU region114 at reset.

SAU_REGION114.LADDR**Type**

int

Default value

0x0

Limit address of SAU region114 at reset.

SAU_REGION114.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region114 at reset.

SAU_REGION115.BADDR**Type**

int

Default value

0x0

Base address of SAU region115 at reset.

SAU_REGION115.ENABLE**Type**

bool

Default value

0x0

Enable SAU region115 at reset.

SAU_REGION115.LADDR**Type**

int

Default value

0x0

Limit address of SAU region115 at reset.

SAU_REGION115.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region115 at reset.

SAU_REGION116.BADDR**Type**

int

Default value

0x0

Base address of SAU region116 at reset.

SAU_REGION116.ENABLE**Type**

bool

Default value

0x0

Enable SAU region116 at reset.

SAU_REGION116.LADDR**Type**

int

Default value

0x0

Limit address of SAU region116 at reset.

SAU_REGION116.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region116 at reset.

SAU_REGION117.BADDR**Type**

int

Default value

0x0

Base address of SAU region117 at reset.

SAU_REGION117.ENABLE**Type**

bool

Default value

0x0

Enable SAU region117 at reset.

SAU_REGION117.LADDR**Type**

int

Default value

0x0

Limit address of SAU region117 at reset.

SAU_REGION117.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region117 at reset.

SAU_REGION118.BADDR**Type**

int

Default value

0x0

Base address of SAU region118 at reset.

SAU_REGION118.ENABLE**Type**

bool

Default value

0x0

Enable SAU region118 at reset.

SAU_REGION118.LADDR**Type**

int

Default value

0x0

Limit address of SAU region118 at reset.

SAU_REGION118.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region118 at reset.

SAU_REGION119.BADDR**Type**

int

Default value

0x0

Base address of SAU region119 at reset.

SAU_REGION119.ENABLE**Type**

bool

Default value

0x0

Enable SAU region119 at reset.

SAU_REGION119.LADDR**Type**

int

Default value

0x0

Limit address of SAU region119 at reset.

SAU_REGION119.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region119 at reset.

SAU_REGION12.BADDR**Type**

int

Default value

0x0

Base address of SAU region12 at reset.

SAU_REGION12.ENABLE**Type**

bool

Default value

0x0

Enable SAU region12 at reset.

SAU_REGION12.LADDR**Type**

int

Default value

0x0

Limit address of SAU region12 at reset.

SAU_REGION12.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region12 at reset.

SAU_REGION120.BADDR**Type**

int

Default value

0x0

Base address of SAU region120 at reset.

SAU_REGION120.ENABLE**Type**

bool

Default value

0x0

Enable SAU region120 at reset.

SAU_REGION120.LADDR**Type**

int

Default value

0x0

Limit address of SAU region120 at reset.

SAU_REGION120.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region120 at reset.

SAU_REGION121.BADDR**Type**

int

Default value

0x0

Base address of SAU region121 at reset.

SAU_REGION121.ENABLE**Type**

bool

Default value

0x0

Enable SAU region121 at reset.

SAU_REGION121.LADDR**Type**

int

Default value

0x0

Limit address of SAU region121 at reset.

SAU_REGION121.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region121 at reset.

SAU_REGION122.BADDR**Type**

int

Default value

0x0

Base address of SAU region122 at reset.

SAU_REGION122.ENABLE**Type**

bool

Default value

0x0

Enable SAU region122 at reset.

SAU_REGION122.LADDR**Type**

int

Default value

0x0

Limit address of SAU region122 at reset.

SAU_REGION122.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region122 at reset.

SAU_REGION123.BADDR**Type**

int

Default value

0x0

Base address of SAU region123 at reset.

SAU_REGION123.ENABLE**Type**

bool

Default value

0x0

Enable SAU region123 at reset.

SAU_REGION123.LADDR**Type**

int

Default value

0x0

Limit address of SAU region123 at reset.

SAU_REGION123.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region123 at reset.

SAU_REGION124.BADDR**Type**

int

Default value

0x0

Base address of SAU region124 at reset.

SAU_REGION124.ENABLE**Type**

bool

Default value

0x0

Enable SAU region124 at reset.

SAU_REGION124.LADDR**Type**

int

Default value

0x0

Limit address of SAU region124 at reset.

SAU_REGION124.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region124 at reset.

SAU_REGION125.BADDR**Type**

int

Default value

0x0

Base address of SAU region125 at reset.

SAU_REGION125.ENABLE**Type**

bool

Default value

0x0

Enable SAU region125 at reset.

SAU_REGION125.LADDR**Type**

int

Default value

0x0

Limit address of SAU region125 at reset.

SAU_REGION125.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region125 at reset.

SAU_REGION126.BADDR**Type**

int

Default value

0x0

Base address of SAU region126 at reset.

SAU_REGION126.ENABLE**Type**

bool

Default value

0x0

Enable SAU region126 at reset.

SAU_REGION126.LADDR**Type**

int

Default value

0x0

Limit address of SAU region126 at reset.

SAU_REGION126.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region126 at reset.

SAU_REGION127.BADDR**Type**

int

Default value

0x0

Base address of SAU region127 at reset.

SAU_REGION127.ENABLE**Type**

bool

Default value

0x0

Enable SAU region127 at reset.

SAU_REGION127.LADDR**Type**

int

Default value

0x0

Limit address of SAU region127 at reset.

SAU_REGION127.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region127 at reset.

SAU_REGION128.BADDR**Type**

int

Default value

0x0

Base address of SAU region128 at reset.

SAU_REGION128.ENABLE**Type**

bool

Default value

0x0

Enable SAU region128 at reset.

SAU_REGION128.LADDR**Type**

int

Default value

0x0

Limit address of SAU region128 at reset.

SAU_REGION128.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region128 at reset.

SAU_REGION129.BADDR**Type**

int

Default value

0x0

Base address of SAU region129 at reset.

SAU_REGION129.ENABLE**Type**

bool

Default value

0x0

Enable SAU region129 at reset.

SAU_REGION129.LADDR**Type**

int

Default value

0x0

Limit address of SAU region129 at reset.

SAU_REGION129.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region129 at reset.

SAU_REGION13.BADDR**Type**

int

Default value

0x0

Base address of SAU region13 at reset.

SAU_REGION13.ENABLE**Type**

bool

Default value

0x0

Enable SAU region13 at reset.

SAU_REGION13.LADDR**Type**

int

Default value

0x0

Limit address of SAU region13 at reset.

SAU_REGION13.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region13 at reset.

SAU_REGION130.BADDR**Type**

int

Default value

0x0

Base address of SAU region130 at reset.

SAU_REGION130.ENABLE**Type**

bool

Default value

0x0

Enable SAU region130 at reset.

SAU_REGION130.LADDR**Type**

int

Default value

0x0

Limit address of SAU region130 at reset.

SAU_REGION130.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region130 at reset.

SAU_REGION131.BADDR**Type**

int

Default value

0x0

Base address of SAU region131 at reset.

SAU_REGION131.ENABLE**Type**

bool

Default value

0x0

Enable SAU region131 at reset.

SAU_REGION131.LADDR**Type**

int

Default value

0x0

Limit address of SAU region131 at reset.

SAU_REGION131.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region131 at reset.

SAU_REGION132.BADDR**Type**

int

Default value

0x0

Base address of SAU region132 at reset.

SAU_REGION132.ENABLE**Type**

bool

Default value

0x0

Enable SAU region132 at reset.

SAU_REGION132.LADDR**Type**

int

Default value

0x0

Limit address of SAU region132 at reset.

SAU_REGION132.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region132 at reset.

SAU_REGION133.BADDR**Type**

int

Default value

0x0

Base address of SAU region133 at reset.

SAU_REGION133.ENABLE**Type**

bool

Default value

0x0

Enable SAU region133 at reset.

SAU_REGION133.LADDR**Type**

int

Default value

0x0

Limit address of SAU region133 at reset.

SAU_REGION133.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region133 at reset.

SAU_REGION134.BADDR**Type**

int

Default value

0x0

Base address of SAU region134 at reset.

SAU_REGION134.ENABLE**Type**

bool

Default value

0x0

Enable SAU region134 at reset.

SAU_REGION134.LADDR**Type**

int

Default value

0x0

Limit address of SAU region134 at reset.

SAU_REGION134.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region134 at reset.

SAU_REGION135.BADDR**Type**

int

Default value

0x0

Base address of SAU region135 at reset.

SAU_REGION135.ENABLE**Type**

bool

Default value

0x0

Enable SAU region135 at reset.

SAU_REGION135.LADDR**Type**

int

Default value

0x0

Limit address of SAU region135 at reset.

SAU_REGION135.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region135 at reset.

SAU_REGION136.BADDR**Type**

int

Default value

0x0

Base address of SAU region136 at reset.

SAU_REGION136.ENABLE**Type**

bool

Default value

0x0

Enable SAU region136 at reset.

SAU_REGION136.LADDR**Type**

int

Default value

0x0

Limit address of SAU region136 at reset.

SAU_REGION136.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region136 at reset.

SAU_REGION137.BADDR**Type**

int

Default value

0x0

Base address of SAU region137 at reset.

SAU_REGION137.ENABLE**Type**

bool

Default value

0x0

Enable SAU region137 at reset.

SAU_REGION137.LADDR**Type**

int

Default value

0x0

Limit address of SAU region137 at reset.

SAU_REGION137.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region137 at reset.

SAU_REGION138.BADDR**Type**

int

Default value

0x0

Base address of SAU region138 at reset.

SAU_REGION138.ENABLE**Type**

bool

Default value

0x0

Enable SAU region138 at reset.

SAU_REGION138.LADDR**Type**

int

Default value

0x0

Limit address of SAU region138 at reset.

SAU_REGION138.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region138 at reset.

SAU_REGION139.BADDR**Type**

int

Default value

0x0

Base address of SAU region139 at reset.

SAU_REGION139.ENABLE**Type**

bool

Default value

0x0

Enable SAU region139 at reset.

SAU_REGION139.LADDR**Type**

int

Default value

0x0

Limit address of SAU region139 at reset.

SAU_REGION139.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region139 at reset.

SAU_REGION14.BADDR**Type**

int

Default value

0x0

Base address of SAU region14 at reset.

SAU_REGION14.ENABLE**Type**

bool

Default value

0x0

Enable SAU region14 at reset.

SAU_REGION14.LADDR**Type**

int

Default value

0x0

Limit address of SAU region14 at reset.

SAU_REGION14.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region14 at reset.

SAU_REGION140.BADDR**Type**

int

Default value

0x0

Base address of SAU region140 at reset.

SAU_REGION140.ENABLE**Type**

bool

Default value

0x0

Enable SAU region140 at reset.

SAU_REGION140.LADDR**Type**

int

Default value

0x0

Limit address of SAU region140 at reset.

SAU_REGION140.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region140 at reset.

SAU_REGION141.BADDR**Type**

int

Default value

0x0

Base address of SAU region141 at reset.

SAU_REGION141.ENABLE**Type**

bool

Default value

0x0

Enable SAU region141 at reset.

SAU_REGION141.LADDR**Type**

int

Default value

0x0

Limit address of SAU region141 at reset.

SAU_REGION141.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region141 at reset.

SAU_REGION142.BADDR**Type**

int

Default value

0x0

Base address of SAU region142 at reset.

SAU_REGION142.ENABLE**Type**

bool

Default value

0x0

Enable SAU region142 at reset.

SAU_REGION142.LADDR**Type**

int

Default value

0x0

Limit address of SAU region142 at reset.

SAU_REGION142.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region142 at reset.

SAU_REGION143.BADDR**Type**

int

Default value

0x0

Base address of SAU region143 at reset.

SAU_REGION143.ENABLE**Type**

bool

Default value

0x0

Enable SAU region143 at reset.

SAU_REGION143.LADDR**Type**

int

Default value

0x0

Limit address of SAU region143 at reset.

SAU_REGION143.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region143 at reset.

SAU_REGION144.BADDR**Type**

int

Default value

0x0

Base address of SAU region144 at reset.

SAU_REGION144.ENABLE**Type**

bool

Default value

0x0

Enable SAU region144 at reset.

SAU_REGION144.LADDR**Type**

int

Default value

0x0

Limit address of SAU region144 at reset.

SAU_REGION144.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region144 at reset.

SAU_REGION145.BADDR**Type**

int

Default value

0x0

Base address of SAU region145 at reset.

SAU_REGION145.ENABLE**Type**

bool

Default value

0x0

Enable SAU region145 at reset.

SAU_REGION145.LADDR**Type**

int

Default value

0x0

Limit address of SAU region145 at reset.

SAU_REGION145.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region145 at reset.

SAU_REGION146.BADDR**Type**

int

Default value

0x0

Base address of SAU region146 at reset.

SAU_REGION146.ENABLE**Type**

bool

Default value

0x0

Enable SAU region146 at reset.

SAU_REGION146.LADDR**Type**

int

Default value

0x0

Limit address of SAU region146 at reset.

SAU_REGION146.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region146 at reset.

SAU_REGION147.BADDR**Type**

int

Default value

0x0

Base address of SAU region147 at reset.

SAU_REGION147.ENABLE**Type**

bool

Default value

0x0

Enable SAU region147 at reset.

SAU_REGION147.LADDR**Type**

int

Default value

0x0

Limit address of SAU region147 at reset.

SAU_REGION147.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region147 at reset.

SAU_REGION148.BADDR**Type**

int

Default value

0x0

Base address of SAU region148 at reset.

SAU_REGION148.ENABLE**Type**

bool

Default value

0x0

Enable SAU region148 at reset.

SAU_REGION148.LADDR**Type**

int

Default value

0x0

Limit address of SAU region148 at reset.

SAU_REGION148.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region148 at reset.

SAU_REGION149.BADDR**Type**

int

Default value

0x0

Base address of SAU region149 at reset.

SAU_REGION149.ENABLE**Type**

bool

Default value

0x0

Enable SAU region149 at reset.

SAU_REGION149.LADDR**Type**

int

Default value

0x0

Limit address of SAU region149 at reset.

SAU_REGION149.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region149 at reset.

SAU_REGION15.BADDR**Type**

int

Default value

0x0

Base address of SAU region15 at reset.

SAU_REGION15.ENABLE**Type**

bool

Default value

0x0

Enable SAU region15 at reset.

SAU_REGION15.LADDR**Type**

int

Default value

0x0

Limit address of SAU region15 at reset.

SAU_REGION15.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region15 at reset.

SAU_REGION150.BADDR**Type**

int

Default value

0x0

Base address of SAU region150 at reset.

SAU_REGION150.ENABLE**Type**

bool

Default value

0x0

Enable SAU region150 at reset.

SAU_REGION150.LADDR**Type**

int

Default value

0x0

Limit address of SAU region150 at reset.

SAU_REGION150.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region150 at reset.

SAU_REGION151.BADDR**Type**

int

Default value

0x0

Base address of SAU region151 at reset.

SAU_REGION151.ENABLE**Type**

bool

Default value

0x0

Enable SAU region151 at reset.

SAU_REGION151.LADDR**Type**

int

Default value

0x0

Limit address of SAU region151 at reset.

SAU_REGION151.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region151 at reset.

SAU_REGION152.BADDR**Type**

int

Default value

0x0

Base address of SAU region152 at reset.

SAU_REGION152.ENABLE**Type**

bool

Default value

0x0

Enable SAU region152 at reset.

SAU_REGION152.LADDR**Type**

int

Default value

0x0

Limit address of SAU region152 at reset.

SAU_REGION152.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region152 at reset.

SAU_REGION153.BADDR**Type**

int

Default value

0x0

Base address of SAU region153 at reset.

SAU_REGION153.ENABLE**Type**

bool

Default value

0x0

Enable SAU region153 at reset.

SAU_REGION153.LADDR**Type**

int

Default value

0x0

Limit address of SAU region153 at reset.

SAU_REGION153.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region153 at reset.

SAU_REGION154.BADDR**Type**

int

Default value

0x0

Base address of SAU region154 at reset.

SAU_REGION154.ENABLE**Type**

bool

Default value

0x0

Enable SAU region154 at reset.

SAU_REGION154.LADDR**Type**

int

Default value

0x0

Limit address of SAU region154 at reset.

SAU_REGION154.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region154 at reset.

SAU_REGION155.BADDR**Type**

int

Default value

0x0

Base address of SAU region155 at reset.

SAU_REGION155.ENABLE**Type**

bool

Default value

0x0

Enable SAU region155 at reset.

SAU_REGION155.LADDR**Type**

int

Default value

0x0

Limit address of SAU region155 at reset.

SAU_REGION155.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region155 at reset.

SAU_REGION156.BADDR**Type**

int

Default value

0x0

Base address of SAU region156 at reset.

SAU_REGION156.ENABLE**Type**

bool

Default value

0x0

Enable SAU region156 at reset.

SAU_REGION156.LADDR**Type**

int

Default value

0x0

Limit address of SAU region156 at reset.

SAU_REGION156.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region156 at reset.

SAU_REGION157.BADDR**Type**

int

Default value

0x0

Base address of SAU region157 at reset.

SAU_REGION157.ENABLE**Type**

bool

Default value

0x0

Enable SAU region157 at reset.

SAU_REGION157.LADDR**Type**

int

Default value

0x0

Limit address of SAU region157 at reset.

SAU_REGION157.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region157 at reset.

SAU_REGION158.BADDR**Type**

int

Default value

0x0

Base address of SAU region158 at reset.

SAU_REGION158.ENABLE**Type**

bool

Default value

0x0

Enable SAU region158 at reset.

SAU_REGION158.LADDR**Type**

int

Default value

0x0

Limit address of SAU region158 at reset.

SAU_REGION158.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region158 at reset.

SAU_REGION159.BADDR**Type**

int

Default value

0x0

Base address of SAU region159 at reset.

SAU_REGION159.ENABLE**Type**

bool

Default value

0x0

Enable SAU region159 at reset.

SAU_REGION159.LADDR**Type**

int

Default value

0x0

Limit address of SAU region159 at reset.

SAU_REGION159.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region159 at reset.

SAU_REGION16.BADDR**Type**

int

Default value

0x0

Base address of SAU region16 at reset.

SAU_REGION16.ENABLE**Type**

bool

Default value

0x0

Enable SAU region16 at reset.

SAU_REGION16.LADDR**Type**

int

Default value

0x0

Limit address of SAU region16 at reset.

SAU_REGION16.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region16 at reset.

SAU_REGION160.BADDR**Type**

int

Default value

0x0

Base address of SAU region160 at reset.

SAU_REGION160.ENABLE**Type**

bool

Default value

0x0

Enable SAU region160 at reset.

SAU_REGION160.LADDR**Type**

int

Default value

0x0

Limit address of SAU region160 at reset.

SAU_REGION160.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region160 at reset.

SAU_REGION161.BADDR**Type**

int

Default value

0x0

Base address of SAU region161 at reset.

SAU_REGION161.ENABLE**Type**

bool

Default value

0x0

Enable SAU region161 at reset.

SAU_REGION161.LADDR**Type**

int

Default value

0x0

Limit address of SAU region161 at reset.

SAU_REGION161.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region161 at reset.

SAU_REGION162.BADDR**Type**

int

Default value

0x0

Base address of SAU region162 at reset.

SAU_REGION162.ENABLE**Type**

bool

Default value

0x0

Enable SAU region162 at reset.

SAU_REGION162.LADDR**Type**

int

Default value

0x0

Limit address of SAU region162 at reset.

SAU_REGION162.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region162 at reset.

SAU_REGION163.BADDR**Type**

int

Default value

0x0

Base address of SAU region163 at reset.

SAU_REGION163.ENABLE**Type**

bool

Default value

0x0

Enable SAU region163 at reset.

SAU_REGION163.LADDR**Type**

int

Default value

0x0

Limit address of SAU region163 at reset.

SAU_REGION163.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region163 at reset.

SAU_REGION164.BADDR**Type**

int

Default value

0x0

Base address of SAU region164 at reset.

SAU_REGION164.ENABLE**Type**

bool

Default value

0x0

Enable SAU region164 at reset.

SAU_REGION164.LADDR**Type**

int

Default value

0x0

Limit address of SAU region164 at reset.

SAU_REGION164.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region164 at reset.

SAU_REGION165.BADDR**Type**

int

Default value

0x0

Base address of SAU region165 at reset.

SAU_REGION165.ENABLE**Type**

bool

Default value

0x0

Enable SAU region165 at reset.

SAU_REGION165.LADDR**Type**

int

Default value

0x0

Limit address of SAU region165 at reset.

SAU_REGION165.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region165 at reset.

SAU_REGION166.BADDR**Type**

int

Default value

0x0

Base address of SAU region166 at reset.

SAU_REGION166.ENABLE**Type**

bool

Default value

0x0

Enable SAU region166 at reset.

SAU_REGION166.LADDR**Type**

int

Default value

0x0

Limit address of SAU region166 at reset.

SAU_REGION166.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region166 at reset.

SAU_REGION167.BADDR**Type**

int

Default value

0x0

Base address of SAU region167 at reset.

SAU_REGION167.ENABLE**Type**

bool

Default value

0x0

Enable SAU region167 at reset.

SAU_REGION167.LADDR**Type**

int

Default value

0x0

Limit address of SAU region167 at reset.

SAU_REGION167.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region167 at reset.

SAU_REGION168.BADDR**Type**

int

Default value

0x0

Base address of SAU region168 at reset.

SAU_REGION168.ENABLE**Type**

bool

Default value

0x0

Enable SAU region168 at reset.

SAU_REGION168.LADDR**Type**

int

Default value

0x0

Limit address of SAU region168 at reset.

SAU_REGION168.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region168 at reset.

SAU_REGION169.BADDR**Type**

int

Default value

0x0

Base address of SAU region169 at reset.

SAU_REGION169.ENABLE**Type**

bool

Default value

0x0

Enable SAU region169 at reset.

SAU_REGION169.LADDR**Type**

int

Default value

0x0

Limit address of SAU region169 at reset.

SAU_REGION169.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region169 at reset.

SAU_REGION17.BADDR**Type**

int

Default value

0x0

Base address of SAU region17 at reset.

SAU_REGION17.ENABLE**Type**

bool

Default value

0x0

Enable SAU region17 at reset.

SAU_REGION17.LADDR**Type**

int

Default value

0x0

Limit address of SAU region17 at reset.

SAU_REGION17.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region17 at reset.

SAU_REGION170.BADDR**Type**

int

Default value

0x0

Base address of SAU region170 at reset.

SAU_REGION170.ENABLE**Type**

bool

Default value

0x0

Enable SAU region170 at reset.

SAU_REGION170.LADDR**Type**

int

Default value

0x0

Limit address of SAU region170 at reset.

SAU_REGION170.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region170 at reset.

SAU_REGION171.BADDR**Type**

int

Default value

0x0

Base address of SAU region171 at reset.

SAU_REGION171.ENABLE**Type**

bool

Default value

0x0

Enable SAU region171 at reset.

SAU_REGION171.LADDR**Type**

int

Default value

0x0

Limit address of SAU region171 at reset.

SAU_REGION171.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region171 at reset.

SAU_REGION172.BADDR**Type**

int

Default value

0x0

Base address of SAU region172 at reset.

SAU_REGION172.ENABLE**Type**

bool

Default value

0x0

Enable SAU region172 at reset.

SAU_REGION172.LADDR**Type**

int

Default value

0x0

Limit address of SAU region172 at reset.

SAU_REGION172.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region172 at reset.

SAU_REGION173.BADDR**Type**

int

Default value

0x0

Base address of SAU region173 at reset.

SAU_REGION173.ENABLE**Type**

bool

Default value

0x0

Enable SAU region173 at reset.

SAU_REGION173.LADDR**Type**

int

Default value

0x0

Limit address of SAU region173 at reset.

SAU_REGION173.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region173 at reset.

SAU_REGION174.BADDR**Type**

int

Default value

0x0

Base address of SAU region174 at reset.

SAU_REGION174.ENABLE**Type**

bool

Default value

0x0

Enable SAU region174 at reset.

SAU_REGION174.LADDR**Type**

int

Default value

0x0

Limit address of SAU region174 at reset.

SAU_REGION174.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region174 at reset.

SAU_REGION175.BADDR**Type**

int

Default value

0x0

Base address of SAU region175 at reset.

SAU_REGION175.ENABLE**Type**

bool

Default value

0x0

Enable SAU region175 at reset.

SAU_REGION175.LADDR**Type**

int

Default value

0x0

Limit address of SAU region175 at reset.

SAU_REGION175.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region175 at reset.

SAU_REGION176.BADDR**Type**

int

Default value

0x0

Base address of SAU region176 at reset.

SAU_REGION176.ENABLE**Type**

bool

Default value

0x0

Enable SAU region176 at reset.

SAU_REGION176.LADDR**Type**

int

Default value

0x0

Limit address of SAU region176 at reset.

SAU_REGION176.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region176 at reset.

SAU_REGION177.BADDR**Type**

int

Default value

0x0

Base address of SAU region177 at reset.

SAU_REGION177.ENABLE**Type**

bool

Default value

0x0

Enable SAU region177 at reset.

SAU_REGION177.LADDR**Type**

int

Default value

0x0

Limit address of SAU region177 at reset.

SAU_REGION177.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region177 at reset.

SAU_REGION178.BADDR**Type**

int

Default value

0x0

Base address of SAU region178 at reset.

SAU_REGION178.ENABLE**Type**

bool

Default value

0x0

Enable SAU region178 at reset.

SAU_REGION178.LADDR**Type**

int

Default value

0x0

Limit address of SAU region178 at reset.

SAU_REGION178.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region178 at reset.

SAU_REGION179.BADDR**Type**

int

Default value

0x0

Base address of SAU region179 at reset.

SAU_REGION179.ENABLE**Type**

bool

Default value

0x0

Enable SAU region179 at reset.

SAU_REGION179.LADDR**Type**

int

Default value

0x0

Limit address of SAU region179 at reset.

SAU_REGION179.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region179 at reset.

SAU_REGION18.BADDR**Type**

int

Default value

0x0

Base address of SAU region18 at reset.

SAU_REGION18.ENABLE**Type**

bool

Default value

0x0

Enable SAU region18 at reset.

SAU_REGION18.LADDR**Type**

int

Default value

0x0

Limit address of SAU region18 at reset.

SAU_REGION18.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region18 at reset.

SAU_REGION180.BADDR**Type**

int

Default value

0x0

Base address of SAU region180 at reset.

SAU_REGION180.ENABLE**Type**

bool

Default value

0x0

Enable SAU region180 at reset.

SAU_REGION180.LADDR**Type**

int

Default value

0x0

Limit address of SAU region180 at reset.

SAU_REGION180.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region180 at reset.

SAU_REGION181.BADDR**Type**

int

Default value

0x0

Base address of SAU region181 at reset.

SAU_REGION181.ENABLE**Type**

bool

Default value

0x0

Enable SAU region181 at reset.

SAU_REGION181.LADDR**Type**

int

Default value

0x0

Limit address of SAU region181 at reset.

SAU_REGION181.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region181 at reset.

SAU_REGION182.BADDR**Type**

int

Default value

0x0

Base address of SAU region182 at reset.

SAU_REGION182.ENABLE**Type**

bool

Default value

0x0

Enable SAU region182 at reset.

SAU_REGION182.LADDR**Type**

int

Default value

0x0

Limit address of SAU region182 at reset.

SAU_REGION182.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region182 at reset.

SAU_REGION183.BADDR**Type**

int

Default value

0x0

Base address of SAU region183 at reset.

SAU_REGION183.ENABLE**Type**

bool

Default value

0x0

Enable SAU region183 at reset.

SAU_REGION183.LADDR**Type**

int

Default value

0x0

Limit address of SAU region183 at reset.

SAU_REGION183.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region183 at reset.

SAU_REGION184.BADDR**Type**

int

Default value

0x0

Base address of SAU region184 at reset.

SAU_REGION184.ENABLE**Type**

bool

Default value

0x0

Enable SAU region184 at reset.

SAU_REGION184.LADDR**Type**

int

Default value

0x0

Limit address of SAU region184 at reset.

SAU_REGION184.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region184 at reset.

SAU_REGION185.BADDR**Type**

int

Default value

0x0

Base address of SAU region185 at reset.

SAU_REGION185.ENABLE**Type**

bool

Default value

0x0

Enable SAU region185 at reset.

SAU_REGION185.LADDR**Type**

int

Default value

0x0

Limit address of SAU region185 at reset.

SAU_REGION185.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region185 at reset.

SAU_REGION186.BADDR**Type**

int

Default value

0x0

Base address of SAU region186 at reset.

SAU_REGION186.ENABLE**Type**

bool

Default value

0x0

Enable SAU region186 at reset.

SAU_REGION186.LADDR**Type**

int

Default value

0x0

Limit address of SAU region186 at reset.

SAU_REGION186.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region186 at reset.

SAU_REGION187.BADDR**Type**

int

Default value

0x0

Base address of SAU region187 at reset.

SAU_REGION187.ENABLE**Type**

bool

Default value

0x0

Enable SAU region187 at reset.

SAU_REGION187.LADDR**Type**

int

Default value

0x0

Limit address of SAU region187 at reset.

SAU_REGION187.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region187 at reset.

SAU_REGION188.BADDR**Type**

int

Default value

0x0

Base address of SAU region188 at reset.

SAU_REGION188.ENABLE**Type**

bool

Default value

0x0

Enable SAU region188 at reset.

SAU_REGION188.LADDR**Type**

int

Default value

0x0

Limit address of SAU region188 at reset.

SAU_REGION188.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region188 at reset.

SAU_REGION189.BADDR**Type**

int

Default value

0x0

Base address of SAU region189 at reset.

SAU_REGION189.ENABLE**Type**

bool

Default value

0x0

Enable SAU region189 at reset.

SAU_REGION189.LADDR**Type**

int

Default value

0x0

Limit address of SAU region189 at reset.

SAU_REGION189.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region189 at reset.

SAU_REGION19.BADDR**Type**

int

Default value

0x0

Base address of SAU region19 at reset.

SAU_REGION19.ENABLE**Type**

bool

Default value

0x0

Enable SAU region19 at reset.

SAU_REGION19.LADDR**Type**

int

Default value

0x0

Limit address of SAU region19 at reset.

SAU_REGION19.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region19 at reset.

SAU_REGION190.BADDR**Type**

int

Default value

0x0

Base address of SAU region190 at reset.

SAU_REGION190.ENABLE**Type**

bool

Default value

0x0

Enable SAU region190 at reset.

SAU_REGION190.LADDR**Type**

int

Default value

0x0

Limit address of SAU region190 at reset.

SAU_REGION190.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region190 at reset.

SAU_REGION191.BADDR**Type**

int

Default value

0x0

Base address of SAU region191 at reset.

SAU_REGION191.ENABLE**Type**

bool

Default value

0x0

Enable SAU region191 at reset.

SAU_REGION191.LADDR**Type**

int

Default value

0x0

Limit address of SAU region191 at reset.

SAU_REGION191.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region191 at reset.

SAU_REGION192.BADDR**Type**

int

Default value

0x0

Base address of SAU region192 at reset.

SAU_REGION192.ENABLE**Type**

bool

Default value

0x0

Enable SAU region192 at reset.

SAU_REGION192.LADDR**Type**

int

Default value

0x0

Limit address of SAU region192 at reset.

SAU_REGION192.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region192 at reset.

SAU_REGION193.BADDR**Type**

int

Default value

0x0

Base address of SAU region193 at reset.

SAU_REGION193.ENABLE**Type**

bool

Default value

0x0

Enable SAU region193 at reset.

SAU_REGION193.LADDR**Type**

int

Default value

0x0

Limit address of SAU region193 at reset.

SAU_REGION193.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region193 at reset.

SAU_REGION194.BADDR**Type**

int

Default value

0x0

Base address of SAU region194 at reset.

SAU_REGION194.ENABLE**Type**

bool

Default value

0x0

Enable SAU region194 at reset.

SAU_REGION194.LADDR**Type**

int

Default value

0x0

Limit address of SAU region194 at reset.

SAU_REGION194.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region194 at reset.

SAU_REGION195.BADDR**Type**

int

Default value

0x0

Base address of SAU region195 at reset.

SAU_REGION195.ENABLE**Type**

bool

Default value

0x0

Enable SAU region195 at reset.

SAU_REGION195.LADDR**Type**

int

Default value

0x0

Limit address of SAU region195 at reset.

SAU_REGION195.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region195 at reset.

SAU_REGION196.BADDR**Type**

int

Default value

0x0

Base address of SAU region196 at reset.

SAU_REGION196.ENABLE**Type**

bool

Default value

0x0

Enable SAU region196 at reset.

SAU_REGION196.LADDR**Type**

int

Default value

0x0

Limit address of SAU region196 at reset.

SAU_REGION196.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region196 at reset.

SAU_REGION197.BADDR**Type**

int

Default value

0x0

Base address of SAU region197 at reset.

SAU_REGION197.ENABLE**Type**

bool

Default value

0x0

Enable SAU region197 at reset.

SAU_REGION197.LADDR**Type**

int

Default value

0x0

Limit address of SAU region197 at reset.

SAU_REGION197.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region197 at reset.

SAU_REGION198.BADDR**Type**

int

Default value

0x0

Base address of SAU region198 at reset.

SAU_REGION198.ENABLE**Type**

bool

Default value

0x0

Enable SAU region198 at reset.

SAU_REGION198.LADDR**Type**

int

Default value

0x0

Limit address of SAU region198 at reset.

SAU_REGION198.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region198 at reset.

SAU_REGION199.BADDR**Type**

int

Default value

0x0

Base address of SAU region199 at reset.

SAU_REGION199.ENABLE**Type**

bool

Default value

0x0

Enable SAU region199 at reset.

SAU_REGION199.LADDR**Type**

int

Default value

0x0

Limit address of SAU region199 at reset.

SAU_REGION199.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region199 at reset.

SAU_REGION2.BADDR**Type**

int

Default value

0x0

Base address of SAU region2 at reset.

SAU_REGION2.ENABLE**Type**

bool

Default value

0x0

Enable SAU region2 at reset.

SAU_REGION2.LADDR**Type**

int

Default value

0x0

Limit address of SAU region2 at reset.

SAU_REGION2.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region2 at reset.

SAU_REGION20.BADDR**Type**

int

Default value

0x0

Base address of SAU region20 at reset.

SAU_REGION20.ENABLE**Type**

bool

Default value

0x0

Enable SAU region20 at reset.

SAU_REGION20.LADDR**Type**

int

Default value

0x0

Limit address of SAU region20 at reset.

SAU_REGION20.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region20 at reset.

SAU_REGION200.BADDR**Type**

int

Default value

0x0

Base address of SAU region200 at reset.

SAU_REGION200.ENABLE**Type**

bool

Default value

0x0

Enable SAU region200 at reset.

SAU_REGION200.LADDR**Type**

int

Default value

0x0

Limit address of SAU region200 at reset.

SAU_REGION200.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region200 at reset.

SAU_REGION201.BADDR**Type**

int

Default value

0x0

Base address of SAU region201 at reset.

SAU_REGION201.ENABLE**Type**

bool

Default value

0x0

Enable SAU region201 at reset.

SAU_REGION201.LADDR**Type**

int

Default value

0x0

Limit address of SAU region201 at reset.

SAU_REGION201.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region201 at reset.

SAU_REGION202.BADDR**Type**

int

Default value

0x0

Base address of SAU region202 at reset.

SAU_REGION202.ENABLE**Type**

bool

Default value

0x0

Enable SAU region202 at reset.

SAU_REGION202.LADDR**Type**

int

Default value

0x0

Limit address of SAU region202 at reset.

SAU_REGION202.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region202 at reset.

SAU_REGION203.BADDR**Type**

int

Default value

0x0

Base address of SAU region203 at reset.

SAU_REGION203.ENABLE**Type**

bool

Default value

0x0

Enable SAU region203 at reset.

SAU_REGION203.LADDR**Type**

int

Default value

0x0

Limit address of SAU region203 at reset.

SAU_REGION203.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region203 at reset.

SAU_REGION204.BADDR**Type**

int

Default value

0x0

Base address of SAU region204 at reset.

SAU_REGION204.ENABLE**Type**

bool

Default value

0x0

Enable SAU region204 at reset.

SAU_REGION204.LADDR**Type**

int

Default value

0x0

Limit address of SAU region204 at reset.

SAU_REGION204.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region204 at reset.

SAU_REGION205.BADDR**Type**

int

Default value

0x0

Base address of SAU region205 at reset.

SAU_REGION205.ENABLE**Type**

bool

Default value

0x0

Enable SAU region205 at reset.

SAU_REGION205.LADDR**Type**

int

Default value

0x0

Limit address of SAU region205 at reset.

SAU_REGION205.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region205 at reset.

SAU_REGION206.BADDR**Type**

int

Default value

0x0

Base address of SAU region206 at reset.

SAU_REGION206.ENABLE**Type**

bool

Default value

0x0

Enable SAU region206 at reset.

SAU_REGION206.LADDR**Type**

int

Default value

0x0

Limit address of SAU region206 at reset.

SAU_REGION206.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region206 at reset.

SAU_REGION207.BADDR**Type**

int

Default value

0x0

Base address of SAU region207 at reset.

SAU_REGION207.ENABLE**Type**

bool

Default value

0x0

Enable SAU region207 at reset.

SAU_REGION207.LADDR**Type**

int

Default value

0x0

Limit address of SAU region207 at reset.

SAU_REGION207.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region207 at reset.

SAU_REGION208.BADDR**Type**

int

Default value

0x0

Base address of SAU region208 at reset.

SAU_REGION208.ENABLE**Type**

bool

Default value

0x0

Enable SAU region208 at reset.

SAU_REGION208.LADDR**Type**

int

Default value

0x0

Limit address of SAU region208 at reset.

SAU_REGION208.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region208 at reset.

SAU_REGION209.BADDR**Type**

int

Default value

0x0

Base address of SAU region209 at reset.

SAU_REGION209.ENABLE**Type**

bool

Default value

0x0

Enable SAU region209 at reset.

SAU_REGION209.LADDR**Type**

int

Default value

0x0

Limit address of SAU region209 at reset.

SAU_REGION209.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region209 at reset.

SAU_REGION21.BADDR**Type**

int

Default value

0x0

Base address of SAU region21 at reset.

SAU_REGION21.ENABLE**Type**

bool

Default value

0x0

Enable SAU region21 at reset.

SAU_REGION21.LADDR**Type**

int

Default value

0x0

Limit address of SAU region21 at reset.

SAU_REGION21.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region21 at reset.

SAU_REGION210.BADDR**Type**

int

Default value

0x0

Base address of SAU region210 at reset.

SAU_REGION210.ENABLE**Type**

bool

Default value

0x0

Enable SAU region210 at reset.

SAU_REGION210.LADDR**Type**

int

Default value

0x0

Limit address of SAU region210 at reset.

SAU_REGION210.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region210 at reset.

SAU_REGION211.BADDR**Type**

int

Default value

0x0

Base address of SAU region211 at reset.

SAU_REGION211.ENABLE**Type**

bool

Default value

0x0

Enable SAU region211 at reset.

SAU_REGION211.LADDR**Type**

int

Default value

0x0

Limit address of SAU region211 at reset.

SAU_REGION211.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region211 at reset.

SAU_REGION212.BADDR**Type**

int

Default value

0x0

Base address of SAU region212 at reset.

SAU_REGION212.ENABLE**Type**

bool

Default value

0x0

Enable SAU region212 at reset.

SAU_REGION212.LADDR**Type**

int

Default value

0x0

Limit address of SAU region212 at reset.

SAU_REGION212.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region212 at reset.

SAU_REGION213.BADDR**Type**

int

Default value

0x0

Base address of SAU region213 at reset.

SAU_REGION213.ENABLE**Type**

bool

Default value

0x0

Enable SAU region213 at reset.

SAU_REGION213.LADDR**Type**

int

Default value

0x0

Limit address of SAU region213 at reset.

SAU_REGION213.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region213 at reset.

SAU_REGION214.BADDR**Type**

int

Default value

0x0

Base address of SAU region214 at reset.

SAU_REGION214.ENABLE**Type**

bool

Default value

0x0

Enable SAU region214 at reset.

SAU_REGION214.LADDR**Type**

int

Default value

0x0

Limit address of SAU region214 at reset.

SAU_REGION214.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region214 at reset.

SAU_REGION215.BADDR**Type**

int

Default value

0x0

Base address of SAU region215 at reset.

SAU_REGION215.ENABLE**Type**

bool

Default value

0x0

Enable SAU region215 at reset.

SAU_REGION215.LADDR**Type**

int

Default value

0x0

Limit address of SAU region215 at reset.

SAU_REGION215.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region215 at reset.

SAU_REGION216.BADDR**Type**

int

Default value

0x0

Base address of SAU region216 at reset.

SAU_REGION216.ENABLE**Type**

bool

Default value

0x0

Enable SAU region216 at reset.

SAU_REGION216.LADDR**Type**

int

Default value

0x0

Limit address of SAU region216 at reset.

SAU_REGION216.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region216 at reset.

SAU_REGION217.BADDR**Type**

int

Default value

0x0

Base address of SAU region217 at reset.

SAU_REGION217.ENABLE**Type**

bool

Default value

0x0

Enable SAU region217 at reset.

SAU_REGION217.LADDR**Type**

int

Default value

0x0

Limit address of SAU region217 at reset.

SAU_REGION217.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region217 at reset.

SAU_REGION218.BADDR**Type**

int

Default value

0x0

Base address of SAU region218 at reset.

SAU_REGION218.ENABLE**Type**

bool

Default value

0x0

Enable SAU region218 at reset.

SAU_REGION218.LADDR**Type**

int

Default value

0x0

Limit address of SAU region218 at reset.

SAU_REGION218.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region218 at reset.

SAU_REGION219.BADDR**Type**

int

Default value

0x0

Base address of SAU region219 at reset.

SAU_REGION219.ENABLE**Type**

bool

Default value

0x0

Enable SAU region219 at reset.

SAU_REGION219.LADDR**Type**

int

Default value

0x0

Limit address of SAU region219 at reset.

SAU_REGION219.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region219 at reset.

SAU_REGION22.BADDR**Type**

int

Default value

0x0

Base address of SAU region22 at reset.

SAU_REGION22.ENABLE**Type**

bool

Default value

0x0

Enable SAU region22 at reset.

SAU_REGION22.LADDR**Type**

int

Default value

0x0

Limit address of SAU region22 at reset.

SAU_REGION22.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region22 at reset.

SAU_REGION220.BADDR**Type**

int

Default value

0x0

Base address of SAU region220 at reset.

SAU_REGION220.ENABLE**Type**

bool

Default value

0x0

Enable SAU region220 at reset.

SAU_REGION220.LADDR**Type**

int

Default value

0x0

Limit address of SAU region220 at reset.

SAU_REGION220.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region220 at reset.

SAU_REGION221.BADDR**Type**

int

Default value

0x0

Base address of SAU region221 at reset.

SAU_REGION221.ENABLE**Type**

bool

Default value

0x0

Enable SAU region221 at reset.

SAU_REGION221.LADDR**Type**

int

Default value

0x0

Limit address of SAU region221 at reset.

SAU_REGION221.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region221 at reset.

SAU_REGION222.BADDR**Type**

int

Default value

0x0

Base address of SAU region222 at reset.

SAU_REGION222.ENABLE**Type**

bool

Default value

0x0

Enable SAU region222 at reset.

SAU_REGION222.LADDR**Type**

int

Default value

0x0

Limit address of SAU region222 at reset.

SAU_REGION222.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region222 at reset.

SAU_REGION223.BADDR**Type**

int

Default value

0x0

Base address of SAU region223 at reset.

SAU_REGION223.ENABLE**Type**

bool

Default value

0x0

Enable SAU region223 at reset.

SAU_REGION223.LADDR**Type**

int

Default value

0x0

Limit address of SAU region223 at reset.

SAU_REGION223.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region223 at reset.

SAU_REGION224.BADDR**Type**

int

Default value

0x0

Base address of SAU region224 at reset.

SAU_REGION224.ENABLE**Type**

bool

Default value

0x0

Enable SAU region224 at reset.

SAU_REGION224.LADDR**Type**

int

Default value

0x0

Limit address of SAU region224 at reset.

SAU_REGION224.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region224 at reset.

SAU_REGION225.BADDR**Type**

int

Default value

0x0

Base address of SAU region225 at reset.

SAU_REGION225.ENABLE**Type**

bool

Default value

0x0

Enable SAU region225 at reset.

SAU_REGION225.LADDR**Type**

int

Default value

0x0

Limit address of SAU region225 at reset.

SAU_REGION225.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region225 at reset.

SAU_REGION226.BADDR**Type**

int

Default value

0x0

Base address of SAU region226 at reset.

SAU_REGION226.ENABLE**Type**

bool

Default value

0x0

Enable SAU region226 at reset.

SAU_REGION226.LADDR**Type**

int

Default value

0x0

Limit address of SAU region226 at reset.

SAU_REGION226.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region226 at reset.

SAU_REGION227.BADDR**Type**

int

Default value

0x0

Base address of SAU region227 at reset.

SAU_REGION227.ENABLE**Type**

bool

Default value

0x0

Enable SAU region227 at reset.

SAU_REGION227.LADDR**Type**

int

Default value

0x0

Limit address of SAU region227 at reset.

SAU_REGION227.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region227 at reset.

SAU_REGION228.BADDR**Type**

int

Default value

0x0

Base address of SAU region228 at reset.

SAU_REGION228.ENABLE**Type**

bool

Default value

0x0

Enable SAU region228 at reset.

SAU_REGION228.LADDR**Type**

int

Default value

0x0

Limit address of SAU region228 at reset.

SAU_REGION228.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region228 at reset.

SAU_REGION229.BADDR**Type**

int

Default value

0x0

Base address of SAU region229 at reset.

SAU_REGION229.ENABLE**Type**

bool

Default value

0x0

Enable SAU region229 at reset.

SAU_REGION229.LADDR**Type**

int

Default value

0x0

Limit address of SAU region229 at reset.

SAU_REGION229.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region229 at reset.

SAU_REGION23.BADDR**Type**

int

Default value

0x0

Base address of SAU region23 at reset.

SAU_REGION23.ENABLE**Type**

bool

Default value

0x0

Enable SAU region23 at reset.

SAU_REGION23.LADDR**Type**

int

Default value

0x0

Limit address of SAU region23 at reset.

SAU_REGION23.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region23 at reset.

SAU_REGION230.BADDR**Type**

int

Default value

0x0

Base address of SAU region230 at reset.

SAU_REGION230.ENABLE**Type**

bool

Default value

0x0

Enable SAU region230 at reset.

SAU_REGION230.LADDR**Type**

int

Default value

0x0

Limit address of SAU region230 at reset.

SAU_REGION230.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region230 at reset.

SAU_REGION231.BADDR**Type**

int

Default value

0x0

Base address of SAU region231 at reset.

SAU_REGION231.ENABLE**Type**

bool

Default value

0x0

Enable SAU region231 at reset.

SAU_REGION231.LADDR**Type**

int

Default value

0x0

Limit address of SAU region231 at reset.

SAU_REGION231.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region231 at reset.

SAU_REGION232.BADDR**Type**

int

Default value

0x0

Base address of SAU region232 at reset.

SAU_REGION232.ENABLE**Type**

bool

Default value

0x0

Enable SAU region232 at reset.

SAU_REGION232.LADDR**Type**

int

Default value

0x0

Limit address of SAU region232 at reset.

SAU_REGION232.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region232 at reset.

SAU_REGION233.BADDR**Type**

int

Default value

0x0

Base address of SAU region233 at reset.

SAU_REGION233.ENABLE**Type**

bool

Default value

0x0

Enable SAU region233 at reset.

SAU_REGION233.LADDR**Type**

int

Default value

0x0

Limit address of SAU region233 at reset.

SAU_REGION233.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region233 at reset.

SAU_REGION234.BADDR**Type**

int

Default value

0x0

Base address of SAU region234 at reset.

SAU_REGION234.ENABLE**Type**

bool

Default value

0x0

Enable SAU region234 at reset.

SAU_REGION234.LADDR**Type**

int

Default value

0x0

Limit address of SAU region234 at reset.

SAU_REGION234.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region234 at reset.

SAU_REGION235.BADDR**Type**

int

Default value

0x0

Base address of SAU region235 at reset.

SAU_REGION235.ENABLE**Type**

bool

Default value

0x0

Enable SAU region235 at reset.

SAU_REGION235.LADDR**Type**

int

Default value

0x0

Limit address of SAU region235 at reset.

SAU_REGION235.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region235 at reset.

SAU_REGION236.BADDR**Type**

int

Default value

0x0

Base address of SAU region236 at reset.

SAU_REGION236.ENABLE**Type**

bool

Default value

0x0

Enable SAU region236 at reset.

SAU_REGION236.LADDR**Type**

int

Default value

0x0

Limit address of SAU region236 at reset.

SAU_REGION236.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region236 at reset.

SAU_REGION237.BADDR**Type**

int

Default value

0x0

Base address of SAU region237 at reset.

SAU_REGION237.ENABLE**Type**

bool

Default value

0x0

Enable SAU region237 at reset.

SAU_REGION237.LADDR**Type**

int

Default value

0x0

Limit address of SAU region237 at reset.

SAU_REGION237.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region237 at reset.

SAU_REGION238.BADDR**Type**

int

Default value

0x0

Base address of SAU region238 at reset.

SAU_REGION238.ENABLE**Type**

bool

Default value

0x0

Enable SAU region238 at reset.

SAU_REGION238.LADDR**Type**

int

Default value

0x0

Limit address of SAU region238 at reset.

SAU_REGION238.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region238 at reset.

SAU_REGION239.BADDR**Type**

int

Default value

0x0

Base address of SAU region239 at reset.

SAU_REGION239.ENABLE**Type**

bool

Default value

0x0

Enable SAU region239 at reset.

SAU_REGION239.LADDR**Type**

int

Default value

0x0

Limit address of SAU region239 at reset.

SAU_REGION239.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region239 at reset.

SAU_REGION24.BADDR**Type**

int

Default value

0x0

Base address of SAU region24 at reset.

SAU_REGION24.ENABLE**Type**

bool

Default value

0x0

Enable SAU region24 at reset.

SAU_REGION24.LADDR**Type**

int

Default value

0x0

Limit address of SAU region24 at reset.

SAU_REGION24.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region24 at reset.

SAU_REGION240.BADDR**Type**

int

Default value

0x0

Base address of SAU region240 at reset.

SAU_REGION240.ENABLE**Type**

bool

Default value

0x0

Enable SAU region240 at reset.

SAU_REGION240.LADDR**Type**

int

Default value

0x0

Limit address of SAU region240 at reset.

SAU_REGION240.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region240 at reset.

SAU_REGION241.BADDR**Type**

int

Default value

0x0

Base address of SAU region241 at reset.

SAU_REGION241.ENABLE**Type**

bool

Default value

0x0

Enable SAU region241 at reset.

SAU_REGION241.LADDR**Type**

int

Default value

0x0

Limit address of SAU region241 at reset.

SAU_REGION241.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region241 at reset.

SAU_REGION242.BADDR**Type**

int

Default value

0x0

Base address of SAU region242 at reset.

SAU_REGION242.ENABLE**Type**

bool

Default value

0x0

Enable SAU region242 at reset.

SAU_REGION242.LADDR**Type**

int

Default value

0x0

Limit address of SAU region242 at reset.

SAU_REGION242.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region242 at reset.

SAU_REGION243.BADDR**Type**

int

Default value

0x0

Base address of SAU region243 at reset.

SAU_REGION243.ENABLE**Type**

bool

Default value

0x0

Enable SAU region243 at reset.

SAU_REGION243.LADDR**Type**

int

Default value

0x0

Limit address of SAU region243 at reset.

SAU_REGION243.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region243 at reset.

SAU_REGION244.BADDR**Type**

int

Default value

0x0

Base address of SAU region244 at reset.

SAU_REGION244.ENABLE**Type**

bool

Default value

0x0

Enable SAU region244 at reset.

SAU_REGION244.LADDR**Type**

int

Default value

0x0

Limit address of SAU region244 at reset.

SAU_REGION244.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region244 at reset.

SAU_REGION245.BADDR**Type**

int

Default value

0x0

Base address of SAU region245 at reset.

SAU_REGION245.ENABLE**Type**

bool

Default value

0x0

Enable SAU region245 at reset.

SAU_REGION245.LADDR**Type**

int

Default value

0x0

Limit address of SAU region245 at reset.

SAU_REGION245.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region245 at reset.

SAU_REGION246.BADDR**Type**

int

Default value

0x0

Base address of SAU region246 at reset.

SAU_REGION246.ENABLE**Type**

bool

Default value

0x0

Enable SAU region246 at reset.

SAU_REGION246.LADDR**Type**

int

Default value

0x0

Limit address of SAU region246 at reset.

SAU_REGION246.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region246 at reset.

SAU_REGION247.BADDR**Type**

int

Default value

0x0

Base address of SAU region247 at reset.

SAU_REGION247.ENABLE**Type**

bool

Default value

0x0

Enable SAU region247 at reset.

SAU_REGION247.LADDR**Type**

int

Default value

0x0

Limit address of SAU region247 at reset.

SAU_REGION247.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region247 at reset.

SAU_REGION248.BADDR**Type**

int

Default value

0x0

Base address of SAU region248 at reset.

SAU_REGION248.ENABLE**Type**

bool

Default value

0x0

Enable SAU region248 at reset.

SAU_REGION248.LADDR**Type**

int

Default value

0x0

Limit address of SAU region248 at reset.

SAU_REGION248.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region248 at reset.

SAU_REGION249.BADDR**Type**

int

Default value

0x0

Base address of SAU region249 at reset.

SAU_REGION249.ENABLE**Type**

bool

Default value

0x0

Enable SAU region249 at reset.

SAU_REGION249.LADDR**Type**

int

Default value

0x0

Limit address of SAU region249 at reset.

SAU_REGION249.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region249 at reset.

SAU_REGION25.BADDR**Type**

int

Default value

0x0

Base address of SAU region25 at reset.

SAU_REGION25.ENABLE**Type**

bool

Default value

0x0

Enable SAU region25 at reset.

SAU_REGION25.LADDR**Type**

int

Default value

0x0

Limit address of SAU region25 at reset.

SAU_REGION25.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region25 at reset.

SAU_REGION250.BADDR**Type**

int

Default value

0x0

Base address of SAU region250 at reset.

SAU_REGION250.ENABLE**Type**

bool

Default value

0x0

Enable SAU region250 at reset.

SAU_REGION250.LADDR**Type**

int

Default value

0x0

Limit address of SAU region250 at reset.

SAU_REGION250.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region250 at reset.

SAU_REGION251.BADDR**Type**

int

Default value

0x0

Base address of SAU region251 at reset.

SAU_REGION251.ENABLE**Type**

bool

Default value

0x0

Enable SAU region251 at reset.

SAU_REGION251.LADDR**Type**

int

Default value

0x0

Limit address of SAU region251 at reset.

SAU_REGION251.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region251 at reset.

SAU_REGION252.BADDR**Type**

int

Default value

0x0

Base address of SAU region252 at reset.

SAU_REGION252.ENABLE**Type**

bool

Default value

0x0

Enable SAU region252 at reset.

SAU_REGION252.LADDR**Type**

int

Default value

0x0

Limit address of SAU region252 at reset.

SAU_REGION252.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region252 at reset.

SAU_REGION253.BADDR**Type**

int

Default value

0x0

Base address of SAU region253 at reset.

SAU_REGION253.ENABLE**Type**

bool

Default value

0x0

Enable SAU region253 at reset.

SAU_REGION253.LADDR**Type**

int

Default value

0x0

Limit address of SAU region253 at reset.

SAU_REGION253.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region253 at reset.

SAU_REGION254.BADDR**Type**

int

Default value

0x0

Base address of SAU region254 at reset.

SAU_REGION254.ENABLE**Type**

bool

Default value

0x0

Enable SAU region254 at reset.

SAU_REGION254.LADDR**Type**

int

Default value

0x0

Limit address of SAU region254 at reset.

SAU_REGION254.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region254 at reset.

SAU_REGION255.BADDR**Type**

int

Default value

0x0

Base address of SAU region255 at reset.

SAU_REGION255.ENABLE**Type**

bool

Default value

0x0

Enable SAU region255 at reset.

SAU_REGION255.LADDR**Type**

int

Default value

0x0

Limit address of SAU region255 at reset.

SAU_REGION255.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region255 at reset.

SAU_REGION26.BADDR**Type**

int

Default value

0x0

Base address of SAU region26 at reset.

SAU_REGION26.ENABLE**Type**

bool

Default value

0x0

Enable SAU region26 at reset.

SAU_REGION26.LADDR**Type**

int

Default value

0x0

Limit address of SAU region26 at reset.

SAU_REGION26.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region26 at reset.

SAU_REGION27.BADDR**Type**

int

Default value

0x0

Base address of SAU region27 at reset.

SAU_REGION27.ENABLE**Type**

bool

Default value

0x0

Enable SAU region27 at reset.

SAU_REGION27.LADDR**Type**

int

Default value

0x0

Limit address of SAU region27 at reset.

SAU_REGION27.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region27 at reset.

SAU_REGION28.BADDR**Type**

int

Default value

0x0

Base address of SAU region28 at reset.

SAU_REGION28.ENABLE**Type**

bool

Default value

0x0

Enable SAU region28 at reset.

SAU_REGION28.LADDR**Type**

int

Default value

0x0

Limit address of SAU region28 at reset.

SAU_REGION28.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region28 at reset.

SAU_REGION29.BADDR**Type**

int

Default value

0x0

Base address of SAU region29 at reset.

SAU_REGION29.ENABLE**Type**

bool

Default value

0x0

Enable SAU region29 at reset.

SAU_REGION29.LADDR**Type**

int

Default value

0x0

Limit address of SAU region29 at reset.

SAU_REGION29.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region29 at reset.

SAU_REGION3.BADDR**Type**

int

Default value

0x0

Base address of SAU region3 at reset.

SAU_REGION3.ENABLE**Type**

bool

Default value

0x0

Enable SAU region3 at reset.

SAU_REGION3.LADDR**Type**

int

Default value

0x0

Limit address of SAU region3 at reset.

SAU_REGION3.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region3 at reset.

SAU_REGION30.BADDR**Type**

int

Default value

0x0

Base address of SAU region30 at reset.

SAU_REGION30.ENABLE**Type**

bool

Default value

0x0

Enable SAU region30 at reset.

SAU_REGION30.LADDR**Type**

int

Default value

0x0

Limit address of SAU region30 at reset.

SAU_REGION30.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region30 at reset.

SAU_REGION31.BADDR**Type**

int

Default value

0x0

Base address of SAU region31 at reset.

SAU_REGION31.ENABLE**Type**

bool

Default value

0x0

Enable SAU region31 at reset.

SAU_REGION31.LADDR**Type**

int

Default value

0x0

Limit address of SAU region31 at reset.

SAU_REGION31.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region31 at reset.

SAU_REGION32.BADDR**Type**

int

Default value

0x0

Base address of SAU region32 at reset.

SAU_REGION32.ENABLE**Type**

bool

Default value

0x0

Enable SAU region32 at reset.

SAU_REGION32.LADDR**Type**

int

Default value

0x0

Limit address of SAU region32 at reset.

SAU_REGION32.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region32 at reset.

SAU_REGION33.BADDR**Type**

int

Default value

0x0

Base address of SAU region33 at reset.

SAU_REGION33.ENABLE**Type**

bool

Default value

0x0

Enable SAU region33 at reset.

SAU_REGION33.LADDR**Type**

int

Default value

0x0

Limit address of SAU region33 at reset.

SAU_REGION33.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region33 at reset.

SAU_REGION34.BADDR**Type**

int

Default value

0x0

Base address of SAU region34 at reset.

SAU_REGION34.ENABLE**Type**

bool

Default value

0x0

Enable SAU region34 at reset.

SAU_REGION34.LADDR**Type**

int

Default value

0x0

Limit address of SAU region34 at reset.

SAU_REGION34.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region34 at reset.

SAU_REGION35.BADDR**Type**

int

Default value

0x0

Base address of SAU region35 at reset.

SAU_REGION35.ENABLE**Type**

bool

Default value

0x0

Enable SAU region35 at reset.

SAU_REGION35.LADDR**Type**

int

Default value

0x0

Limit address of SAU region35 at reset.

SAU_REGION35.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region35 at reset.

SAU_REGION36.BADDR**Type**

int

Default value

0x0

Base address of SAU region36 at reset.

SAU_REGION36.ENABLE**Type**

bool

Default value

0x0

Enable SAU region36 at reset.

SAU_REGION36.LADDR**Type**

int

Default value

0x0

Limit address of SAU region36 at reset.

SAU_REGION36.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region36 at reset.

SAU_REGION37.BADDR**Type**

int

Default value

0x0

Base address of SAU region37 at reset.

SAU_REGION37.ENABLE**Type**

bool

Default value

0x0

Enable SAU region37 at reset.

SAU_REGION37.LADDR**Type**

int

Default value

0x0

Limit address of SAU region37 at reset.

SAU_REGION37.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region37 at reset.

SAU_REGION38.BADDR**Type**

int

Default value

0x0

Base address of SAU region38 at reset.

SAU_REGION38.ENABLE**Type**

bool

Default value

0x0

Enable SAU region38 at reset.

SAU_REGION38.LADDR**Type**

int

Default value

0x0

Limit address of SAU region38 at reset.

SAU_REGION38.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region38 at reset.

SAU_REGION39.BADDR**Type**

int

Default value

0x0

Base address of SAU region39 at reset.

SAU_REGION39.ENABLE**Type**

bool

Default value

0x0

Enable SAU region39 at reset.

SAU_REGION39.LADDR**Type**

int

Default value

0x0

Limit address of SAU region39 at reset.

SAU_REGION39.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region39 at reset.

SAU_REGION4.BADDR**Type**

int

Default value

0x0

Base address of SAU region4 at reset.

SAU_REGION4.ENABLE**Type**

bool

Default value

0x0

Enable SAU region4 at reset.

SAU_REGION4.LADDR**Type**

int

Default value

0x0

Limit address of SAU region4 at reset.

SAU_REGION4.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region4 at reset.

SAU_REGION40.BADDR**Type**

int

Default value

0x0

Base address of SAU region40 at reset.

SAU_REGION40.ENABLE**Type**

bool

Default value

0x0

Enable SAU region40 at reset.

SAU_REGION40.LADDR**Type**

int

Default value

0x0

Limit address of SAU region40 at reset.

SAU_REGION40.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region40 at reset.

SAU_REGION41.BADDR**Type**

int

Default value

0x0

Base address of SAU region41 at reset.

SAU_REGION41.ENABLE**Type**

bool

Default value

0x0

Enable SAU region41 at reset.

SAU_REGION41.LADDR**Type**

int

Default value

0x0

Limit address of SAU region41 at reset.

SAU_REGION41.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region41 at reset.

SAU_REGION42.BADDR**Type**

int

Default value

0x0

Base address of SAU region42 at reset.

SAU_REGION42.ENABLE**Type**

bool

Default value

0x0

Enable SAU region42 at reset.

SAU_REGION42.LADDR**Type**

int

Default value

0x0

Limit address of SAU region42 at reset.

SAU_REGION42.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region42 at reset.

SAU_REGION43.BADDR**Type**

int

Default value

0x0

Base address of SAU region43 at reset.

SAU_REGION43.ENABLE**Type**

bool

Default value

0x0

Enable SAU region43 at reset.

SAU_REGION43.LADDR**Type**

int

Default value

0x0

Limit address of SAU region43 at reset.

SAU_REGION43.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region43 at reset.

SAU_REGION44.BADDR**Type**

int

Default value

0x0

Base address of SAU region44 at reset.

SAU_REGION44.ENABLE**Type**

bool

Default value

0x0

Enable SAU region44 at reset.

SAU_REGION44.LADDR**Type**

int

Default value

0x0

Limit address of SAU region44 at reset.

SAU_REGION44.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region44 at reset.

SAU_REGION45.BADDR**Type**

int

Default value

0x0

Base address of SAU region45 at reset.

SAU_REGION45.ENABLE**Type**

bool

Default value

0x0

Enable SAU region45 at reset.

SAU_REGION45.LADDR**Type**

int

Default value

0x0

Limit address of SAU region45 at reset.

SAU_REGION45.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region45 at reset.

SAU_REGION46.BADDR**Type**

int

Default value

0x0

Base address of SAU region46 at reset.

SAU_REGION46.ENABLE**Type**

bool

Default value

0x0

Enable SAU region46 at reset.

SAU_REGION46.LADDR**Type**

int

Default value

0x0

Limit address of SAU region46 at reset.

SAU_REGION46.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region46 at reset.

SAU_REGION47.BADDR**Type**

int

Default value

0x0

Base address of SAU region47 at reset.

SAU_REGION47.ENABLE**Type**

bool

Default value

0x0

Enable SAU region47 at reset.

SAU_REGION47.LADDR**Type**

int

Default value

0x0

Limit address of SAU region47 at reset.

SAU_REGION47.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region47 at reset.

SAU_REGION48.BADDR**Type**

int

Default value

0x0

Base address of SAU region48 at reset.

SAU_REGION48.ENABLE**Type**

bool

Default value

0x0

Enable SAU region48 at reset.

SAU_REGION48.LADDR**Type**

int

Default value

0x0

Limit address of SAU region48 at reset.

SAU_REGION48.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region48 at reset.

SAU_REGION49.BADDR**Type**

int

Default value

0x0

Base address of SAU region49 at reset.

SAU_REGION49.ENABLE**Type**

bool

Default value

0x0

Enable SAU region49 at reset.

SAU_REGION49.LADDR**Type**

int

Default value

0x0

Limit address of SAU region49 at reset.

SAU_REGION49.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region49 at reset.

SAU_REGION5.BADDR**Type**

int

Default value

0x0

Base address of SAU region5 at reset.

SAU_REGION5.ENABLE**Type**

bool

Default value

0x0

Enable SAU region5 at reset.

SAU_REGION5.LADDR**Type**

int

Default value

0x0

Limit address of SAU region5 at reset.

SAU_REGION5.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region5 at reset.

SAU_REGION50.BADDR**Type**

int

Default value

0x0

Base address of SAU region50 at reset.

SAU_REGION50.ENABLE**Type**

bool

Default value

0x0

Enable SAU region50 at reset.

SAU_REGION50.LADDR**Type**

int

Default value

0x0

Limit address of SAU region50 at reset.

SAU_REGION50.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region50 at reset.

SAU_REGION51.BADDR**Type**

int

Default value

0x0

Base address of SAU region51 at reset.

SAU_REGION51.ENABLE**Type**

bool

Default value

0x0

Enable SAU region51 at reset.

SAU_REGION51.LADDR**Type**

int

Default value

0x0

Limit address of SAU region51 at reset.

SAU_REGION51.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region51 at reset.

SAU_REGION52.BADDR**Type**

int

Default value

0x0

Base address of SAU region52 at reset.

SAU_REGION52.ENABLE**Type**

bool

Default value

0x0

Enable SAU region52 at reset.

SAU_REGION52.LADDR**Type**

int

Default value

0x0

Limit address of SAU region52 at reset.

SAU_REGION52.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region52 at reset.

SAU_REGION53.BADDR**Type**

int

Default value

0x0

Base address of SAU region53 at reset.

SAU_REGION53.ENABLE**Type**

bool

Default value

0x0

Enable SAU region53 at reset.

SAU_REGION53.LADDR**Type**

int

Default value

0x0

Limit address of SAU region53 at reset.

SAU_REGION53.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region53 at reset.

SAU_REGION54.BADDR**Type**

int

Default value

0x0

Base address of SAU region54 at reset.

SAU_REGION54.ENABLE**Type**

bool

Default value

0x0

Enable SAU region54 at reset.

SAU_REGION54.LADDR**Type**

int

Default value

0x0

Limit address of SAU region54 at reset.

SAU_REGION54.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region54 at reset.

SAU_REGION55.BADDR**Type**

int

Default value

0x0

Base address of SAU region55 at reset.

SAU_REGION55.ENABLE**Type**

bool

Default value

0x0

Enable SAU region55 at reset.

SAU_REGION55.LADDR**Type**

int

Default value

0x0

Limit address of SAU region55 at reset.

SAU_REGION55.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region55 at reset.

SAU_REGION56.BADDR**Type**

int

Default value

0x0

Base address of SAU region56 at reset.

SAU_REGION56.ENABLE**Type**

bool

Default value

0x0

Enable SAU region56 at reset.

SAU_REGION56.LADDR**Type**

int

Default value

0x0

Limit address of SAU region56 at reset.

SAU_REGION56.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region56 at reset.

SAU_REGION57.BADDR**Type**

int

Default value

0x0

Base address of SAU region57 at reset.

SAU_REGION57.ENABLE**Type**

bool

Default value

0x0

Enable SAU region57 at reset.

SAU_REGION57.LADDR**Type**

int

Default value

0x0

Limit address of SAU region57 at reset.

SAU_REGION57.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region57 at reset.

SAU_REGION58.BADDR**Type**

int

Default value

0x0

Base address of SAU region58 at reset.

SAU_REGION58.ENABLE**Type**

bool

Default value

0x0

Enable SAU region58 at reset.

SAU_REGION58.LADDR**Type**

int

Default value

0x0

Limit address of SAU region58 at reset.

SAU_REGION58.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region58 at reset.

SAU_REGION59.BADDR**Type**

int

Default value

0x0

Base address of SAU region59 at reset.

SAU_REGION59.ENABLE**Type**

bool

Default value

0x0

Enable SAU region59 at reset.

SAU_REGION59.LADDR**Type**

int

Default value

0x0

Limit address of SAU region59 at reset.

SAU_REGION59.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region59 at reset.

SAU_REGION6.BADDR**Type**

int

Default value

0x0

Base address of SAU region6 at reset.

SAU_REGION6.ENABLE**Type**

bool

Default value

0x0

Enable SAU region6 at reset.

SAU_REGION6.LADDR**Type**

int

Default value

0x0

Limit address of SAU region6 at reset.

SAU_REGION6.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region6 at reset.

SAU_REGION60.BADDR**Type**

int

Default value

0x0

Base address of SAU region60 at reset.

SAU_REGION60.ENABLE**Type**

bool

Default value

0x0

Enable SAU region60 at reset.

SAU_REGION60.LADDR**Type**

int

Default value

0x0

Limit address of SAU region60 at reset.

SAU_REGION60.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region60 at reset.

SAU_REGION61.BADDR**Type**

int

Default value

0x0

Base address of SAU region61 at reset.

SAU_REGION61.ENABLE**Type**

bool

Default value

0x0

Enable SAU region61 at reset.

SAU_REGION61.LADDR**Type**

int

Default value

0x0

Limit address of SAU region61 at reset.

SAU_REGION61.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region61 at reset.

SAU_REGION62.BADDR**Type**

int

Default value

0x0

Base address of SAU region62 at reset.

SAU_REGION62.ENABLE**Type**

bool

Default value

0x0

Enable SAU region62 at reset.

SAU_REGION62.LADDR**Type**

int

Default value

0x0

Limit address of SAU region62 at reset.

SAU_REGION62.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region62 at reset.

SAU_REGION63.BADDR**Type**

int

Default value

0x0

Base address of SAU region63 at reset.

SAU_REGION63.ENABLE**Type**

bool

Default value

0x0

Enable SAU region63 at reset.

SAU_REGION63.LADDR**Type**

int

Default value

0x0

Limit address of SAU region63 at reset.

SAU_REGION63.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region63 at reset.

SAU_REGION64.BADDR**Type**

int

Default value

0x0

Base address of SAU region64 at reset.

SAU_REGION64.ENABLE**Type**

bool

Default value

0x0

Enable SAU region64 at reset.

SAU_REGION64.LADDR**Type**

int

Default value

0x0

Limit address of SAU region64 at reset.

SAU_REGION64.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region64 at reset.

SAU_REGION65.BADDR**Type**

int

Default value

0x0

Base address of SAU region65 at reset.

SAU_REGION65.ENABLE**Type**

bool

Default value

0x0

Enable SAU region65 at reset.

SAU_REGION65.LADDR**Type**

int

Default value

0x0

Limit address of SAU region65 at reset.

SAU_REGION65.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region65 at reset.

SAU_REGION66.BADDR**Type**

int

Default value

0x0

Base address of SAU region66 at reset.

SAU_REGION66.ENABLE**Type**

bool

Default value

0x0

Enable SAU region66 at reset.

SAU_REGION66.LADDR**Type**

int

Default value

0x0

Limit address of SAU region66 at reset.

SAU_REGION66.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region66 at reset.

SAU_REGION67.BADDR**Type**

int

Default value

0x0

Base address of SAU region67 at reset.

SAU_REGION67.ENABLE**Type**

bool

Default value

0x0

Enable SAU region67 at reset.

SAU_REGION67.LADDR**Type**

int

Default value

0x0

Limit address of SAU region67 at reset.

SAU_REGION67.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region67 at reset.

SAU_REGION68.BADDR**Type**

int

Default value

0x0

Base address of SAU region68 at reset.

SAU_REGION68.ENABLE**Type**

bool

Default value

0x0

Enable SAU region68 at reset.

SAU_REGION68.LADDR**Type**

int

Default value

0x0

Limit address of SAU region68 at reset.

SAU_REGION68.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region68 at reset.

SAU_REGION69.BADDR**Type**

int

Default value

0x0

Base address of SAU region69 at reset.

SAU_REGION69.ENABLE**Type**

bool

Default value

0x0

Enable SAU region69 at reset.

SAU_REGION69.LADDR**Type**

int

Default value

0x0

Limit address of SAU region69 at reset.

SAU_REGION69.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region69 at reset.

SAU_REGION7.BADDR**Type**

int

Default value

0x0

Base address of SAU region7 at reset.

SAU_REGION7.ENABLE**Type**

bool

Default value

0x0

Enable SAU region7 at reset.

SAU_REGION7.LADDR**Type**

int

Default value

0x0

Limit address of SAU region7 at reset.

SAU_REGION7.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region7 at reset.

SAU_REGION70.BADDR**Type**

int

Default value

0x0

Base address of SAU region70 at reset.

SAU_REGION70.ENABLE**Type**

bool

Default value

0x0

Enable SAU region70 at reset.

SAU_REGION70.LADDR**Type**

int

Default value

0x0

Limit address of SAU region70 at reset.

SAU_REGION70.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region70 at reset.

SAU_REGION71.BADDR**Type**

int

Default value

0x0

Base address of SAU region71 at reset.

SAU_REGION71.ENABLE**Type**

bool

Default value

0x0

Enable SAU region71 at reset.

SAU_REGION71.LADDR**Type**

int

Default value

0x0

Limit address of SAU region71 at reset.

SAU_REGION71.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region71 at reset.

SAU_REGION72.BADDR**Type**

int

Default value

0x0

Base address of SAU region72 at reset.

SAU_REGION72.ENABLE**Type**

bool

Default value

0x0

Enable SAU region72 at reset.

SAU_REGION72.LADDR**Type**

int

Default value

0x0

Limit address of SAU region72 at reset.

SAU_REGION72.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region72 at reset.

SAU_REGION73.BADDR**Type**

int

Default value

0x0

Base address of SAU region73 at reset.

SAU_REGION73.ENABLE**Type**

bool

Default value

0x0

Enable SAU region73 at reset.

SAU_REGION73.LADDR**Type**

int

Default value

0x0

Limit address of SAU region73 at reset.

SAU_REGION73.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region73 at reset.

SAU_REGION74.BADDR**Type**

int

Default value

0x0

Base address of SAU region74 at reset.

SAU_REGION74.ENABLE**Type**

bool

Default value

0x0

Enable SAU region74 at reset.

SAU_REGION74.LADDR**Type**

int

Default value

0x0

Limit address of SAU region74 at reset.

SAU_REGION74.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region74 at reset.

SAU_REGION75.BADDR**Type**

int

Default value

0x0

Base address of SAU region75 at reset.

SAU_REGION75.ENABLE**Type**

bool

Default value

0x0

Enable SAU region75 at reset.

SAU_REGION75.LADDR**Type**

int

Default value

0x0

Limit address of SAU region75 at reset.

SAU_REGION75.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region75 at reset.

SAU_REGION76.BADDR**Type**

int

Default value

0x0

Base address of SAU region76 at reset.

SAU_REGION76.ENABLE**Type**

bool

Default value

0x0

Enable SAU region76 at reset.

SAU_REGION76.LADDR**Type**

int

Default value

0x0

Limit address of SAU region76 at reset.

SAU_REGION76.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region76 at reset.

SAU_REGION77.BADDR**Type**

int

Default value

0x0

Base address of SAU region77 at reset.

SAU_REGION77.ENABLE**Type**

bool

Default value

0x0

Enable SAU region77 at reset.

SAU_REGION77.LADDR**Type**

int

Default value

0x0

Limit address of SAU region77 at reset.

SAU_REGION77.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region77 at reset.

SAU_REGION78.BADDR**Type**

int

Default value

0x0

Base address of SAU region78 at reset.

SAU_REGION78.ENABLE**Type**

bool

Default value

0x0

Enable SAU region78 at reset.

SAU_REGION78.LADDR**Type**

int

Default value

0x0

Limit address of SAU region78 at reset.

SAU_REGION78.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region78 at reset.

SAU_REGION79.BADDR**Type**

int

Default value

0x0

Base address of SAU region79 at reset.

SAU_REGION79.ENABLE**Type**

bool

Default value

0x0

Enable SAU region79 at reset.

SAU_REGION79.LADDR**Type**

int

Default value

0x0

Limit address of SAU region79 at reset.

SAU_REGION79.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region79 at reset.

SAU_REGION8.BADDR**Type**

int

Default value

0x0

Base address of SAU region8 at reset.

SAU_REGION8.ENABLE**Type**

bool

Default value

0x0

Enable SAU region8 at reset.

SAU_REGION8.LADDR**Type**

int

Default value

0x0

Limit address of SAU region8 at reset.

SAU_REGION8.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region8 at reset.

SAU_REGION80.BADDR**Type**

int

Default value

0x0

Base address of SAU region80 at reset.

SAU_REGION80.ENABLE**Type**

bool

Default value

0x0

Enable SAU region80 at reset.

SAU_REGION80.LADDR**Type**

int

Default value

0x0

Limit address of SAU region80 at reset.

SAU_REGION80.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region80 at reset.

SAU_REGION81.BADDR**Type**

int

Default value

0x0

Base address of SAU region81 at reset.

SAU_REGION81.ENABLE**Type**

bool

Default value

0x0

Enable SAU region81 at reset.

SAU_REGION81.LADDR**Type**

int

Default value

0x0

Limit address of SAU region81 at reset.

SAU_REGION81.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region81 at reset.

SAU_REGION82.BADDR**Type**

int

Default value

0x0

Base address of SAU region82 at reset.

SAU_REGION82.ENABLE**Type**

bool

Default value

0x0

Enable SAU region82 at reset.

SAU_REGION82.LADDR**Type**

int

Default value

0x0

Limit address of SAU region82 at reset.

SAU_REGION82.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region82 at reset.

SAU_REGION83.BADDR**Type**

int

Default value

0x0

Base address of SAU region83 at reset.

SAU_REGION83.ENABLE**Type**

bool

Default value

0x0

Enable SAU region83 at reset.

SAU_REGION83.LADDR**Type**

int

Default value

0x0

Limit address of SAU region83 at reset.

SAU_REGION83.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region83 at reset.

SAU_REGION84.BADDR**Type**

int

Default value

0x0

Base address of SAU region84 at reset.

SAU_REGION84.ENABLE**Type**

bool

Default value

0x0

Enable SAU region84 at reset.

SAU_REGION84.LADDR**Type**

int

Default value

0x0

Limit address of SAU region84 at reset.

SAU_REGION84.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region84 at reset.

SAU_REGION85.BADDR**Type**

int

Default value

0x0

Base address of SAU region85 at reset.

SAU_REGION85.ENABLE**Type**

bool

Default value

0x0

Enable SAU region85 at reset.

SAU_REGION85.LADDR**Type**

int

Default value

0x0

Limit address of SAU region85 at reset.

SAU_REGION85.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region85 at reset.

SAU_REGION86.BADDR**Type**

int

Default value

0x0

Base address of SAU region86 at reset.

SAU_REGION86.ENABLE**Type**

bool

Default value

0x0

Enable SAU region86 at reset.

SAU_REGION86.LADDR**Type**

int

Default value

0x0

Limit address of SAU region86 at reset.

SAU_REGION86.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region86 at reset.

SAU_REGION87.BADDR**Type**

int

Default value

0x0

Base address of SAU region87 at reset.

SAU_REGION87.ENABLE**Type**

bool

Default value

0x0

Enable SAU region87 at reset.

SAU_REGION87.LADDR**Type**

int

Default value

0x0

Limit address of SAU region87 at reset.

SAU_REGION87.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region87 at reset.

SAU_REGION88.BADDR**Type**

int

Default value

0x0

Base address of SAU region88 at reset.

SAU_REGION88.ENABLE**Type**

bool

Default value

0x0

Enable SAU region88 at reset.

SAU_REGION88.LADDR**Type**

int

Default value

0x0

Limit address of SAU region88 at reset.

SAU_REGION88.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region88 at reset.

SAU_REGION89.BADDR**Type**

int

Default value

0x0

Base address of SAU region89 at reset.

SAU_REGION89.ENABLE**Type**

bool

Default value

0x0

Enable SAU region89 at reset.

SAU_REGION89.LADDR**Type**

int

Default value

0x0

Limit address of SAU region89 at reset.

SAU_REGION89.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region89 at reset.

SAU_REGION9.BADDR**Type**

int

Default value

0x0

Base address of SAU region9 at reset.

SAU_REGION9.ENABLE**Type**

bool

Default value

0x0

Enable SAU region9 at reset.

SAU_REGION9.LADDR**Type**

int

Default value

0x0

Limit address of SAU region9 at reset.

SAU_REGION9.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region9 at reset.

SAU_REGION90.BADDR**Type**

int

Default value

0x0

Base address of SAU region90 at reset.

SAU_REGION90.ENABLE**Type**

bool

Default value

0x0

Enable SAU region90 at reset.

SAU_REGION90.LADDR**Type**

int

Default value

0x0

Limit address of SAU region90 at reset.

SAU_REGION90.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region90 at reset.

SAU_REGION91.BADDR**Type**

int

Default value

0x0

Base address of SAU region91 at reset.

SAU_REGION91.ENABLE**Type**

bool

Default value

0x0

Enable SAU region91 at reset.

SAU_REGION91.LADDR**Type**

int

Default value

0x0

Limit address of SAU region91 at reset.

SAU_REGION91.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region91 at reset.

SAU_REGION92.BADDR**Type**

int

Default value

0x0

Base address of SAU region92 at reset.

SAU_REGION92.ENABLE**Type**

bool

Default value

0x0

Enable SAU region92 at reset.

SAU_REGION92.LADDR**Type**

int

Default value

0x0

Limit address of SAU region92 at reset.

SAU_REGION92.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region92 at reset.

SAU_REGION93.BADDR**Type**

int

Default value

0x0

Base address of SAU region93 at reset.

SAU_REGION93.ENABLE**Type**

bool

Default value

0x0

Enable SAU region93 at reset.

SAU_REGION93.LADDR**Type**

int

Default value

0x0

Limit address of SAU region93 at reset.

SAU_REGION93.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region93 at reset.

SAU_REGION94.BADDR**Type**

int

Default value

0x0

Base address of SAU region94 at reset.

SAU_REGION94.ENABLE**Type**

bool

Default value

0x0

Enable SAU region94 at reset.

SAU_REGION94.LADDR**Type**

int

Default value

0x0

Limit address of SAU region94 at reset.

SAU_REGION94.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region94 at reset.

SAU_REGION95.BADDR**Type**

int

Default value

0x0

Base address of SAU region95 at reset.

SAU_REGION95.ENABLE**Type**

bool

Default value

0x0

Enable SAU region95 at reset.

SAU_REGION95.LADDR**Type**

int

Default value

0x0

Limit address of SAU region95 at reset.

SAU_REGION95.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region95 at reset.

SAU_REGION96.BADDR**Type**

int

Default value

0x0

Base address of SAU region96 at reset.

SAU_REGION96.ENABLE**Type**

bool

Default value

0x0

Enable SAU region96 at reset.

SAU_REGION96.LADDR**Type**

int

Default value

0x0

Limit address of SAU region96 at reset.

SAU_REGION96.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region96 at reset.

SAU_REGION97.BADDR**Type**

int

Default value

0x0

Base address of SAU region97 at reset.

SAU_REGION97.ENABLE**Type**

bool

Default value

0x0

Enable SAU region97 at reset.

SAU_REGION97.LADDR**Type**

int

Default value

0x0

Limit address of SAU region97 at reset.

SAU_REGION97.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region97 at reset.

SAU_REGION98.BADDR**Type**

int

Default value

0x0

Base address of SAU region98 at reset.

SAU_REGION98.ENABLE**Type**

bool

Default value

0x0

Enable SAU region98 at reset.

SAU_REGION98.LADDR**Type**

int

Default value

0x0

Limit address of SAU region98 at reset.

SAU_REGION98.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region98 at reset.

SAU_REGION99.BADDR**Type**

int

Default value

0x0

Base address of SAU region99 at reset.

SAU_REGION99.ENABLE**Type**

bool

Default value

0x0

Enable SAU region99 at reset.

SAU_REGION99.LADDR**Type**

int

Default value

0x0

Limit address of SAU region99 at reset.

SAU_REGION99.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region99 at reset.

SAU_TYPE.SREGION**Type**

int

Default value

0x10

Number of SAU regions (0 => no SAU).

SECEXT**Type**

bool

Default value

0x1

Whether the ARMv8-M Security Extensions are included.

SYST**Type**

int

Default value

0x2

Include SysTick timer functionality (0=Absent, 1=Secure only, 2=Secure and NS).

SYST_CALIB_NS_reset**Type**

int

Default value

0x0

SYST_CALIB_NS reset value.

SYST_CALIB_reset**Type**

int

Default value

0x0

SYST_CALIB reset value.

VTOR_NS**Type**

bool

Default value

0x1

NonSecure Vector Table Offset Register is writeable.

VTOR_NS_MASK**Type**

int

Default value

0xffffffff80

Non-Secure VTOR write mask.

VTOR_S**Type**

bool

Default value

0x1

Secure Vector Table Offset Register is writeable.

VTOR_S_MASK**Type**

int

Default value

0xffffffff80

Secure VTOR write mask.

WIC**Type**

bool

Default value

0x1

Include support for WIC-mode deep sleep.

abort_unaligned_nonNormal**Type**

bool

Default value

0x1

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

IS the AIRCR.IESB bit [5] writable?.

aircr_iesb_reset**Type**

bool

Default value

0x0

Set the AIRCR.IESB bit [5] after reset.

allow_dap_writes_while_core_running**Type**

bool

Default value

0x1

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

Allow handling Debug Monitor exception with in-flight instructions.

allow_stack_accesses_to_ppb_space**Type**

bool

Default value

0x0

Allow stack accesses to PPB space.

always_undefinstr_over_nocp**Type**

bool

Default value

0x0

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

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

Use the baseline profile (if false, use mainline).

bp_on_2nd_halfword**Type**

bool

Default value

0x1

Respect DWT/BPU breakpoint-hit on 2nd halfword of 32-bit instruction.

callee_register_push_low_to_high**Type**

bool

Default value

0x1

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

Run FP checks on both supported and unsupported CDE instructions.

cde_impl_name**Type**

string

Default value

""

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

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

Reset Value of condition flags in APSR.

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

cpu_can_access_debug_regs**Type**

bool

Default value

0x1

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

Enable LDCX and STCX instructions.

dcache-invalidate-ns-cleans-s**Type**

bool

Default value

0x0

Whether V8M DCI* in non-secure should clean-and-invalidate secure cache contents.

dcache-size**Type**

int

Default value

0x8000

L1 D-cache size in bytes.

dcache-state_modelled**Type**

bool

Default value

0x0

Set whether D-cache has stateful implementation.

delay_faultmask_update**Type**

bool

Default value

0x0

Delay FAULTMASK update to context sync.

delay_sysreg_update**Type**

bool

Default value

0x0

Delay some system register updates (e.g. SHCSR) to context sync.

do_exclusive_monitor_check_first**Type**

bool

Default value

0x0

In exclusive stores, check local exclusive monitor before detecting other memory aborts.

drop_mem_fault**Type**

bool

Default value

0x0

Whether to drop mem_fault in favour of subsequently generated NOCP/secure fault in PushStack.

dtcm_enable**Type**

bool

Default value

0x0

Enable DTCM at reset.

dtcm_size**Type**

int

Default value

0x100

DTCM size in KB.

dwt_unaligned_word_access_as_half_word**Type**

bool

Default value

0x1

DWT Treat unaligned word access as half word or bytes.

enable_helium_extension**Type**

bool

Default value

0x0

Enable Helium extension.

exercise_strex_fail**Type**

bool

Default value

0x0

Reject a pseudo-random majority of exclusive store instructions.

has_ahbp**Type**

bool

Default value

0x1

Are Vendor-Sys accesses sent to a separate bus (AHBP on CM7).

has_arm_v8-1m**Type**

bool

Default value

0x0

Enable v8.1M architecture version and features.

has_cde**Type**

bool

Default value

0x0

Enables Custom Datapath Extensions.

has_core_dside_bus_gasket**Type**

bool

Default value

0x0

STL gasket enabled.

has_lob_cache**Type**

bool

Default value

0x1

Support for LOB cache (only if support for LO instructions is enabled as well).

has_m55_tcmcr**Type**

bool

Default value

0x0

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

Availability of optional PMU.

has_separate_etm_reset**Type**

bool

Default value

0x0

If true, signal 'etmreset' resets the core, else the core power-on-reset does.

has_unpriviledged_debug**Type**

bool

Default value

0x1

Unprivileged Debug Extension supported for Mainline Extension.

has_writebuffer**Type**

bool

Default value

0x0

Implement write accesses buffering before L1 cache. May affect ext_abort behaviour.

icache-size**Type**

int

Default value

0x8000

L1 I-cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

ignore_RNR_top_nibble**Type**

bool

Default value

0x0

If set, only the bottom four bits of MPU_RNR.REGION are used.

ignore_demcr_sdme_for_nonhalting_bkpt**Type**

bool

Default value

0x0

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

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

Use smaller decoder does not UNDEF some unpredictable SBZ/SBO fields.

ignore_unpred_ZeroRegistersInList**Type**

bool

Default value

0x0

VLDM,VSTM,STM,LDM with no registers NOP instead of UNDEF.

itcm_enable**Type**

bool

Default value

0x0

Enable ITCM at reset.

itcm_size**Type**

int

Default value

0x100

ITCM size in KB.

late_arrival**Type**

bool

Default value

0x1

Enable late arrival support.

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

mve_has_atomic_ticks**Type**

bool

Default value

0x0

Enable atomic ticks behaviour for vector instructions flagged as such (e.g. VLDR).

num_pmu_counters**Type**

int

Default value

0x1f

Number of available PMU counters.

number_of_itm_stimulus_ports**Type**

int

Default value

0x20

The number of ITM stimulus ports.

pend_overriden_exception_on_stack_push**Type**

bool

Default value

0x0

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

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}"

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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}"
```

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

RAS: Critical error interrupt pin.

ras_cei_support**Type**

bool

Default value

0x1

RAS: Whether Critical Error Interrupt is supported.

ras_eri_pin**Type**

int

Default value

0x1

RAS: Error recovery interrupt pin.

ras_eri_support**Type**

bool

Default value

0x1

RAS: Whether Error Recovery Interrupt is supported.

ras_error_record**Type**

int

Default value

0xffffffffffffffff

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

RAS: Fault handling interrupt pin.

ras_fhi_support**Type**

bool

Default value

0x1

RAS: Whether Fault Handling Interrupt is supported.

rd_ns_bus_err_behave**Type**

int

Default value

0x1

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

External read aborts in secure domain 0:ignored, 1:precise, 2:imprecise, 3=imprecise except SO.

register_reset_data**Type**

int

Default value

0x0

Data used to fill register bits when they become UNKNOWN at reset.

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit**Type**

int

Default value

0x20700000

Virtual address of top of heap.

semihosting-prefix**Type**

bool

Default value

0x0

Prefix semihosting output with target instance name.

semihosting-stack_base**Type**

int

Default value

0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit**Type**

int

Default value

0x20700000

Virtual address of stack limit.

sep_sec_state_then_apply_prigroup_to_pending**Type**

bool

Default value

0x0

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

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

If true, Fault Address Register is shared.

stack_limit_check**Type**

int

Default value

0x7

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

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

Faults on stack writes are precise.

supports_unprivileged

Type

bool

Default value

0x1

Enable support for Unprivileged/Privileged Extension.

tail_chain

Type

bool

Default value

0x1

Enable tail-chaining optimisation.

trace_style

Type

int

Default value

0x2

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

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

allow STM with write back to base register in register list.

unpred_mon_step_write**Type**

int

Default value

0x0

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

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

If true, MSR to *PSR with a zero mask does nothing.

unstack_R_regs_before_fp_cp_check**Type**

bool

Default value

0x0

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

Watchpoint on exception vector fetch.

vector_fetch_busfault_sets_HFSR_FORCED**Type**

bool

Default value

0x0

Only v8.0M. Set HFSR.FORCED when a vector table read generates a HardFault.

vector_fetch_on_iside**Type**

bool

Default value

0x1

Perform vector fetch on I-side.

vfp-enable_at_reset**Type**

bool

Default value

0x0

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

Set whether the model has VFP support.

wr_ns_bus_err_behave

Type
int

Default value
0x3

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

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

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 3-200: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA5CT

This model has the following Iris instances:

Table 3-201: ARMCortexA5CT Iris instances

InstanceName	ComponentName
ARMCortexA5CT	Cluster_ARM_Cortex-A5UP

InstanceName	ComponentName
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.l1licache	PVCache
ARMCortexA5CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA5CT.cpu0.utlb	TlbCadi
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

This model has the following MTI trace components:

Table 3-202: 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

InstanceName	ComponentName
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

ARMCortexA5CT contains the following CADI targets:

- ARM_Cortex-A5UP
- Cluster_ARM_Cortex-A5UP
- PVCache
- TlbCadi

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.

Ports for ARMCortexA5CT**Table 3-203: 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 ARMCortexA5CT

CLUSTER_ID

Type

int

Default value

0x0

Processor cluster ID value.

cpi_div

Type

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul

Type

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND

Type

bool

Default value

0x0

Initialize to BE8 endianness.

cpuX.CFGNMFI

Type

bool

Default value

0x0

Enable nonmaskable FIQ interrupts on startup.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.POWERCTLI**Type**

int

Default value

0x0

Default power control state for processor.

cpuX.TEINIT**Type**

bool

Default value

0x0

T32 exception enable. The default has exceptions including reset handled in A32 state.

cpuX.VINITHI**Type**

bool

Default value

0x0

Initialize with high vectors enabled.

cpuX.ase-present**Type**

bool

Default value

0x1

Set whether model has NEON support.

cpuX.dcache-size**Type**

int

Default value

0x8000

Set D-cache size in bytes.

cpuX.icache-size**Type**

int

Default value

0x8000

Set I-cache size in bytes.

cpuX.min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-ARM_HLT**Type**

int

Default value

0xf000

ARM HLT number for semihosting.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

ARM SVC number for semihosting.

cpuX.semihosting-Thumb_HLT**Type**

int

Default value

0x3c

Thumb HLT number for semihosting.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

Thumb SVC number for semihosting.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

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

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-hlt-enable**Type**

bool

Default value

0x0

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

Prefix semihosting output with target instance name.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

Enable coprocessor access and VFP at reset.

cpuX.vfp-present**Type**

bool

Default value

0x1

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

Specify whether all TLBs are modeled.

`icache-hit_latency`**Type**

int

Default value

0x0

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

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

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

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

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

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 3-204: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA5MPx1CT

This model has the following Iris instances:

Table 3-205: 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.l1icache	PVCache
ARMCortexA5MPx1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA5MPx1CT.cpu0.utlb	TlbCadi
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

InstanceName	ComponentName
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 3-206: 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
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave

InstanceName	ComponentName
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexA5MPx1CT.l2_flusher	AsyncCacheFlushUnit

ARMCortexA5MPx1CT contains the following CADI targets:

- ARM_Cortex-A5MP
- Cluster_ARM_Cortex-A5MP
- PVCache
- TlbCadi

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 `ARMCortexA5MPxnCT` 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.

Ports for ARMCortexA5MPx1CT

Table 3-207: 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.

Name	Protocol	Type	Description
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.
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

Disable some accesses to GIC registers.

CLUSTER_ID

Type

int

Default value

0x0

Processor cluster ID value.

FILTEREN

Type

bool

Default value

0x0

Enable filtering of accesses through pvbus_m0.

FILTEREND**Type**

int

Default value

0x0

End of region filtered to pvbus_m0.

FILTERSTART**Type**

int

Default value

0x0

Base of region filtered to pvbus_m0.

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Initialize to BE8 endianness.

cpuX.CFGNMFI**Type**

bool

Default value

0x0

Enable nonmaskable FIQ interrupts on startup.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.POWERCTLI**Type**

int

Default value

0x0

Default power control state for processor.

cpuX.SMPnAMP**Type**

bool

Default value

0x0

Set whether the processor is part of a coherent domain.

cpuX.TEINIT**Type**

bool

Default value

0x0

T32 exception enable. The default has exceptions including reset handled in A32 state.

cpuX.VINITHI**Type**

bool

Default value

0x0

Initialize with high vectors enabled.

cpuX.ase-present**Type**

bool

Default value

0x1

Set whether model has NEON support.

cpuX.dcache-size**Type**

int

Default value

0x8000

Set D-cache size in bytes.

cpuX.icache-size**Type**

int

Default value

0x8000

Set I-cache size in bytes.

cpuX.min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-ARM_HLT**Type**

int

Default value

0xf000

ARM HLT number for semihosting.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

ARM SVC number for semihosting.

cpuX.semihosting-Thumb_HLT**Type**

int

Default value

0x3c

Thumb HLT number for semihosting.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

Thumb SVC number for semihosting.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

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

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-hlt-enable**Type**

bool

Default value

0x0

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

Prefix semihosting output with target instance name.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

Enable coprocessor access and VFP at reset.

cpuX.vfp-present**Type**

bool

Default value

0x1

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`

Type

int

Default value

0x0

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

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

Specify whether all TLBs are modeled.

dic-spi_count**Type**

int

Default value

0x40

Number of shared peripheral interrupts implemented.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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 3-208: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA7x1CT

This model has the following Iris instances:

Table 3-209: ARMCortexA7x1CT Iris instances

InstanceName	ComponentName
ARMCortexA7x1CT	Cluster_ARM_Cortex-A7
ARMCortexA7x1CT.Cortex-A7_GIC	GICv2
ARMCortexA7x1CT.acp_mapper	PVBusMapper
ARMCortexA7x1CT.cpu0	ARM_Cortex-A7
ARMCortexA7x1CT.cpu0.DTLB	TLB
ARMCortexA7x1CT.cpu0.ITLB	TLB
ARMCortexA7x1CT.cpu0.dtlb	TlbCadi
ARMCortexA7x1CT.cpu0.itlb	TlbCadi
ARMCortexA7x1CT.cpu0.l1dcache	PVCache
ARMCortexA7x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA7x1CT.cpu0.l1icache	PVCache
ARMCortexA7x1CT.cpu0.l1icache.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

InstanceName	ComponentName
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 3-210: 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.l1icache	PVCache
ARMCortexA7x1CT.cpu0.l1icache.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

InstanceName	ComponentName
ARMCortexA7x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA7x1CT.l2_flusher	AsyncCacheFlushUnit

ARMCortexA7x1CT contains the following CADI targets:

- ARM_Cortex-A7
- Cluster_ARM_Cortex-A7
- PVCache
- TlbCadi

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 ARMCortexA7x_nCT component in a VE platform model, the `PERIPBASE` 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.

Ports for ARMCortexA7x1CT

Table 3-211: 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 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.

Name	Protocol	Type	Description
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.
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 ARMCortexA7x1CT

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTINNER

Type

bool

Default value

0x1

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CFGSDISABLE**Type**

bool

Default value

0x0

Disable some accesses to GIC registers.

CLUSTER_ID**Type**

int

Default value

0x0

Processor cluster ID value.

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

`cpi_mul`**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

`cpuX.CFGEND`**Type**

bool

Default value

0x0

Initialize to BE8 endianness.

`cpuX.CP15SDISABLE`**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

`cpuX.DBGROMADDR`**Type**

int

Default value

0x12000003

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

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

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

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

T32 exception enable. The default has exceptions including reset handled in A32 state.

cpuX.VINITHI**Type**

bool

Default value

0x0

Initialize with high vectors enabled.

cpuX.ase-present**Type**

bool

Default value

0x1

Set whether CT model has been built with NEON support.

cpuX.l1_dcach**Type**

int

Default value

0x8000

Size of L1 D-cache.

cpuX.l1_icach**Type**

int

Default value

0x8000

Size of L1 I-cache.

cpuX.min_sync_level**Type**

int

Default value

0x0

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-ARM_HLT**Type**

int

Default value

0xf000

ARM HLT number for semihosting.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

ARM SVC number for semihosting.

cpuX.semihosting-Thumb_HLT**Type**

int

Default value

0x3c

Thumb HLT number for semihosting.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

Thumb SVC number for semihosting.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

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

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-hlt-enable**Type**

bool

Default value

0x0

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

Prefix semihosting output with target instance name.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

Enable coprocessor access and VFP at reset.

cpuX.vfp-present**Type**

bool

Default value

0x1

Set whether CT model has been built with VFP support.

dic-spi_count**Type**

int

Default value

0x40

Number of shared peripheral interrupts implemented.

disable_periph_decoder**Type**

bool

Default value

0x0

Disable memory mapped access to gic system registers.

internal_vgic**Type**

bool

Default value

0x1

Configures whether the model of the processor contains a VGIC.

l1_dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

Set whether L1 D-cache has stateful implementation.

`l1_dcache-write_access_latency`

Type

int

Default value

0x0

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_dcachelwrite_latency**Type**

int

Default value

0x0

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

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_icachelmaintenance_latency**Type**

int

Default value

0x0

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

l1_icachelmiss_latency**Type**

int

Default value

0x0

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

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

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

Set whether L1 I-cache has stateful implementation.

`l2_cache-hit_latency`

Type

int

Default value

0x0

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

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

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

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

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

Set L2 cache size in bytes.

l2_cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

Set whether L2 cache has stateful implementation.

l2_cache-write_access_latency**Type**

int

Default value

0x0

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

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 3-212: IP revisions support

Revision	Quality level
r2p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA8CT

This model has the following Iris instances:

Table 3-213: ARMCortexA8CT Iris instances

InstanceName	ComponentName
ARMCortexA8CT	ARM_Cortex-A8
ARMCortexA8CT.DTLB	TLB
ARMCortexA8CT.ITLB	TLB
ARMCortexA8CT.acp_mapper	PVBusMapper
ARMCortexA8CT.cpu0.dtlb	TlbCadi
ARMCortexA8CT.cpu0.itlb	TlbCadi
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

InstanceName	ComponentName
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

This model has the following MTI trace components:

Table 3-214: 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

ARMCortexA8CT contains the following CADI targets:

- ARM_Cortex-A8
- PVCache
- TlbCadi

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.

Ports for ARMCortexA8CT

Table 3-215: 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.
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.
pvbush_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 ARMCortexA8CT

CFGEND0

Type

bool

Default value

0x0

Initialize to BE8 endianness.

CFGNMFI

Type

bool

Default value

0x0

Enable nonmaskable FIQ interrupts on startup.

CFGTE

Type

bool

Default value

0x0

Initialize to take exceptions in T32 state. Model starts in T32 state.

CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

VINITHI**Type**

bool

Default value

0x0

Initialize with high vectors enabled.

`cpi_div`**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

`cpi_mul`**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

`device-accurate-tlb`**Type**

bool

Default value

0x0

Specify whether all TLBs are modeled.

implements_vfp**Type**

bool

Default value

0x1

Set whether the model has been built with VFP and NEON support.

l1_dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

Set L1 D-cache size in bytes.

`l1_dcache-state_modelled`

Type

bool

Default value

0x0

Include Level 1 data cache state model.

`l1_dcache-write_access_latency`

Type

int

Default value

0x0

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_dcachelwrite_latency**Type**

int

Default value

0x0

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

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

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

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_icachelread_latency is set. This is in addition to the hit or miss latency, 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_icachelread_latency**Type**

int

Default value

0x0

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

Set L1 I-cache size in bytes.

`l1_icache-state_modelled`**Type**

bool

Default value

0x0

Include Level 1 instruction cache state model.

`l2_cache-hit_latency`**Type**

int

Default value

0x0

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

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

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

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

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

Set L2 cache size in bytes.

l2_cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Include unified Level 2 cache state model.

`l2_cache-write_access_latency`**Type**

int

Default value

0x0

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

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

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

semihosting-ARM_HLT**Type**

int

Default value

0xf000

ARM HLT number for semihosting.

semihosting-ARM_SVC**Type**

int

Default value

0x123456

ARM SVC number for semihosting.

semihosting-Thumb_HLT**Type**

int

Default value

0x3c

Thumb HLT number for semihosting.

semihosting-Thumb_SVC**Type**

int

Default value

0xab

Thumb SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

semihosting-hlt-enable**Type**

bool

Default value

0x0

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

Prefix semihosting output with target instance name.

semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

siliconID**Type**

int

Default value

0x41000000

Value as read by the system coprocessor siliconID register.

vfp-enable_at_reset**Type**

bool

Default value

0x0

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 3-216: IP revisions support

Revision	Quality level
r3p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA9MPx1CT

This model has the following Iris instances:

Table 3-217: ARMCortexA9MPx1CT Iris instances

InstanceName	ComponentName
ARMCortexA9MPx1CT	Cluster_ARM_Cortex-A9MP
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	TlbCadi
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 3-218: 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

ARMCortexA9MPx1CT contains the following CADI targets:

- ARM_Cortex-A9MP
- Cluster_ARM_Cortex-A9MP
- PVCache
- TlbCadi

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_nCT 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.

Ports for ARMCortexA9MPx1CT

Table 3-219: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>cfgend[4]</code>	Signal	Slave	This signal is for EE bit initialisation.

Name	Protocol	Type	Description
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.
pwrctlhi[4]	Value	Slave	This port sets reset value for scu CPU status register.
pwrctllo[4]	Value	Master	This port sends scu CPU status register bits.
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 ARMCortexA9MPx1CT

CFGSDISABLE

Type

bool

Default value

0x0

Disable some accesses to GIC registers.

CLUSTER_ID

Type

int

Default value

0x0

Processor cluster ID value.

FILTEREN

Type

bool

Default value

0x0

Enable filtering of accesses through pvbus_m0.

FILTEREND

Type

int

Default value

0x0

End of region filtered to pvbus_m0.

FILTERSTART

Type

int

Default value

0x0

Base of region filtered to pvbus_m0.

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Initialize to BE8 endianness.

cpuX.CFGNMFI**Type**

bool

Default value

0x0

Enable nonmaskable FIQ interrupts on startup.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.POWERCTLI**Type**

int

Default value

0x0

Default power control state for processor.

cpuX.SMPnAMP**Type**

bool

Default value

0x0

Set whether the processor is part of a coherent domain.

cpuX.TEINIT**Type**

bool

Default value

0x0

T32 exception enable. The default has exceptions including reset handled in A32 state.

cpuX.VINITHI**Type**

bool

Default value

0x0

Initialize with high vectors enabled.

cpuX.ase-present**Type**

bool

Default value

0x1

Set whether model has NEON support.

cpuX.dcache-size**Type**

int

Default value

0x8000

Set D-cache size in bytes.

cpuX.icache-size**Type**

int

Default value

0x8000

Set I-cache size in bytes.

cpuX.min_sync_level**Type**

int

Default value

0x0

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-ARM_HLT**Type**

int

Default value

0xf000

ARM HLT number for semihosting.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

ARM SVC number for semihosting.

cpuX.semihosting-Thumb_HLT**Type**

int

Default value

0x3c

Thumb HLT number for semihosting.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

Thumb SVC number for semihosting.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

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

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-hlt-enable**Type**

bool

Default value

0x0

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

Prefix semihosting output with target instance name.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

Enable coprocessor access and VFP at reset.

cpuX.vfp-present**Type**

bool

Default value

0x1

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`**Type**

int

Default value

0x0

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

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

Specify whether all TLBs are modeled.

dic-spi_count**Type**

int

Default value

0x40

Number of shared peripheral interrupts implemented.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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 3-220: IP revisions support

Revision	Quality level
r3p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA9UPCT

This model has the following Iris instances:

Table 3-221: ARMCortexA9UPCT Iris instances

InstanceName	ComponentName
ARMCortexA9UPCT	Cluster_ARM_Cortex-A9UP
ARMCortexA9UPCT.acp_mapper	PVBusMapper
ARMCortexA9UPCT.cpu0	ARM_Cortex-A9UP
ARMCortexA9UPCT.cpu0.UTLB	TLB

InstanceName	ComponentName
ARMCortexA9UPCT.cpu0.l1dcache	PVCache
ARMCortexA9UPCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA9UPCT.cpu0.l1licache	PVCache
ARMCortexA9UPCT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA9UPCT.cpu0.utlb	TlbCadi
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 3-222: 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

InstanceName	ComponentName
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

ARMCortexA9UPCT contains the following CADI targets:

- ARM_Cortex-A9UP
- Cluster_ARM_Cortex-A9UP
- PVCache
- TlbCadi

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.

Ports for ARMCortexA9UPCT

Table 3-223: Ports

Name	Protocol	Type	Description
<code>cfgend[1]</code>	Signal	Slave	This signal is for EE bit initialisation.
<code>cfgnmfi[1]</code>	Signal	Slave	This signal disables FIQ mask in CPSR.
<code>clk_in</code>	ClockSignal	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the core executes instructions.
<code>clusterid</code>	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.

Name	Protocol	Type	Description
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

Processor cluster ID value.

cpi_div

Type

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul

Type

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Initialize to BE8 endianness.

cpuX.CFGNMFI**Type**

bool

Default value

0x0

Enable nonmaskable FIQ interrupts on startup.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.POWERCTLI**Type**

int

Default value

0x0

Default power control state for processor.

cpuX.TEINIT**Type**

bool

Default value

0x0

T32 exception enable. The default has exceptions including reset handled in A32 state.

cpuX.VINITHI**Type**

bool

Default value

0x0

Initialize with high vectors enabled.

cpuX.ase-present**Type**

bool

Default value

0x1

Set whether model has NEON support.

cpuX.dcache-size**Type**

int

Default value

0x8000

Set D-cache size in bytes.

cpuX.icache-size**Type**

int

Default value

0x8000

Set I-cache size in bytes.

cpuX.min_sync_level**Type**

int

Default value

0x0

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-ARM_HLT**Type**

int

Default value

0xf000

ARM HLT number for semihosting.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

ARM SVC number for semihosting.

cpuX.semihosting-Thumb_HLT**Type**

int

Default value

0x3c

Thumb HLT number for semihosting.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

Thumb SVC number for semihosting.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

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

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-hlt-enable**Type**

bool

Default value

0x0

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

Prefix semihosting output with target instance name.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

Enable coprocessor access and VFP at reset.

cpuX.vfp-present**Type**

bool

Default value

0x1

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`

Type

int

Default value

0x0

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

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

Specify whether all TLBs are modeled.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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 3-224: IP revisions support

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA15x1CT

This model has the following Iris instances:

Table 3-225: ARMCortexA15x1CT Iris instances

InstanceName	ComponentName
ARMCortexA15x1CT	Cluster_ARM_Cortex-A15
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	TlbCadi
ARMCortexA15x1CT.cpu0.itlb	TlbCadi
ARMCortexA15x1CT.cpu0.l1dcache	PVCache
ARMCortexA15x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA15x1CT.cpu0.l1icache	PVCache
ARMCortexA15x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA15x1CT.ext_bus	PVBusLogger

InstanceName	ComponentName
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 3-226: 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.l1icache	PVCache
ARMCortexA15x1CT.cpu0.l1icache.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

InstanceName	ComponentName
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

ARMCortexA15x1CT contains the following CADI targets:

- ARM_Cortex-A15
- Cluster_ARM_Cortex-A15
- PVCache
- TlbCadi

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 ARMCortexA15x_nCT 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.

Ports for ARMCortexA15x1CT

Table 3-227: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>broadcastcachemaint</code>	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
<code>broadcastinner</code>	Signal	Slave	Enable broadcasting of Inner Shareable transactions.
<code>broadcastouter</code>	Signal	Slave	Enable broadcasting of Outer Shareable transactions.

Name	Protocol	Type	Description
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.
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 SCTL.R.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 ARMCortexA15x1CT

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTINNER

Type

bool

Default value

0x1

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CFGSDISABLE

Type

bool

Default value

0x0

Disable some accesses to GIC registers.

CLUSTER_ID

Type

int

Default value

0x0

Processor cluster ID value.

IMINLN**Type**

bool

Default value

0x1

Instruction cache minimum line size: false=32 bytes, true=64 bytes.

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Initialize to BE8 endianness.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.DBGROMADDR**Type**

int

Default value

0x12000003

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

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

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

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

T32 exception enable. The default has exceptions including reset handled in A32 state.

cpuX.VINITHI**Type**

bool

Default value

0x0

Initialize with high vectors enabled.

cpuX.ase-present**Type**

bool

Default value

0x1

Set whether CT model has been built with NEON support.

cpuX.min_sync_level**Type**

int

Default value

0x0

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-ARM_HLT**Type**

int

Default value

0xf000

ARM HLT number for semihosting.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

ARM SVC number for semihosting.

cpuX.semihosting-Thumb_HLT**Type**

int

Default value

0x3c

Thumb HLT number for semihosting.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

Thumb SVC number for semihosting.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

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

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-hlt-enable**Type**

bool

Default value

0x0

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

Prefix semihosting output with target instance name.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

Enable coprocessor access and VFP at reset.

cpuX.vfp-present**Type**

bool

Default value

0x1

Set whether CT model has been built with VFP support.

dic-spi_count**Type**

int

Default value

0x40

Number of shared peripheral interrupts implemented.

disable_periph_decoder**Type**

bool

Default value

0x0

Disable memory mapped access to gic system registers.

internal_vgic**Type**

bool

Default value

0x1

Configures whether the model of the processor contains a Virtualized Generic Interrupt Controller (VGIC).

l1_dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

Set whether L1 D-cache has stateful implementation.

l1_dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

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

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_ichache-read_access_latency**Type**

int

Default value

0x0

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_ichache-read_latency is set. This is in addition to the hit or miss latency, 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-read_latency**Type**

int

Default value

0x0

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

Set whether L1 I-cache has stateful implementation.

l2-data-slice**Type**

int

Default value

0x0

L2 data RAM slice.

l2-tag-slice**Type**

int

Default value

0x0

L2 tag RAM slice.

l2_cache-hit_latency

Type

int

Default value

0x0

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

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

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

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

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

Set L2 cache size in bytes.

`l2_cache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

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

Set whether L2 cache has stateful implementation.

l2_cache-write_access_latency

Type
int

Default value
0x0

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

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 3-228: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA17x1CT

This model has the following Iris instances:

Table 3-229: ARMCortexA17x1CT Iris instances

InstanceName	ComponentName
ARMCortexA17x1CT	Cluster_ARM_Cortex-A17
ARMCortexA17x1CT.acp_mapper	PVBusMapper
ARMCortexA17x1CT.cpu0	ARM_Cortex-A17
ARMCortexA17x1CT.cpu0.DTLB	TLB
ARMCortexA17x1CT.cpu0.ITLB	TLB
ARMCortexA17x1CT.cpu0.dtlb	TlbCadi

InstanceName	ComponentName
ARMCortexA17x1CT.cpu0.itlb	TlbCadi
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
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 3-230: 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

InstanceName	ComponentName
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
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

ARMCortexA17x1CT contains the following CADI targets:

- ARM_Cortex-A17
- Cluster_ARM_Cortex-A17
- PVCache
- TlbCadi

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.

Ports for ARMCortexA17x1CT**Table 3-231: Ports**

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>broadcastcachemaint</code>	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
<code>broadcastinner</code>	Signal	Slave	Enable broadcasting of Inner Shareable transactions.
<code>broadcastouter</code>	Signal	Slave	Enable broadcasting of Outer Shareable transactions.
<code>CFGADDRFILTEENDNS</code>	Value_64	Slave	NS end address (only bits 39:20 are used).
<code>CFGADDRFILTEENDS</code>	Value_64	Slave	S end address (only bits 39:20 are used).
<code>CFGADDRFILTEENNS</code>	Signal	Slave	Enable periph port filtering for NS accesses.
<code>CFGADDRFILTEENS</code>	Signal	Slave	Enable periph port filtering for S accesses.
<code>CFGADDRFILTESTARTNS</code>	Value_64	Slave	NS start address (only bits 39:20 are used).
<code>CFGADDRFILTESTARTS</code>	Value_64	Slave	S start address (only bits 39:20 are used).
<code>cfgend[4]</code>	Signal	Slave	This signal controls the SCTLR.EE bit.
<code>clk_in</code>	ClockSignal	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the core executes instructions.
<code>clusterid</code>	Value	Slave	This port sets the value in the CLUSTERID field (bits[11:8]) of the MPIDR.
<code>CNTHPIRQ[4]</code>	Signal	Master	Outputs of the generic timers.
<code>CNTPNSIRQ[4]</code>	Signal	Master	Outputs of the generic timers.
<code>CNTPSIRQ[4]</code>	Signal	Master	Outputs of the generic timers.
<code>cntvalueb</code>	CounterInterface	Slave	Interface to SoC level counter module.
<code>CNTVIRQ[4]</code>	Signal	Master	Outputs of the generic timers.
<code>cp15sdisable[4]</code>	Signal	Slave	This signal disables write access to some secure system control processor registers.
<code>cpuporeset[4]</code>	Signal	Slave	Signal initializes all processor logic including NEON, VFP, Debug, PTM, breakpoint and watchpoint.
<code>event</code>	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
<code>fiq[4]</code>	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
<code>irq[4]</code>	Signal	Slave	This signal drives the CPU's interrupt handling.
<code>l2flushdone</code>	Signal	Master	Flush of L2 memory system complete.

Name	Protocol	Type	Description
l2flushreq	Signal	Slave	Request flush of L2 memory system.
l2reset	Signal	Slave	This signal resets the shared L2 memory system and timer logic.
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 ARMCortexA17x1CT

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTINNER

Type

bool

Default value

0x1

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CFGADDRFILTEENDNS**Type**

int

Default value

0x0

Peripheral port NS end address.

CFGADDRFILTEENDS**Type**

int

Default value

0x0

Peripheral port S end address.

CFGADDRFILTEENNS**Type**

bool

Default value

0x0

Peripheral port NS address filtering enabled.

CFGADDRFILTEENS**Type**

bool

Default value

0x0

Peripheral port S address filtering enabled.

CFGADDRFILTSTARTNS**Type**

int

Default value

0x0

Peripheral port NS start address.

CFGADDRFILTSTARTS**Type**

int

Default value

0x0

Peripheral port S start address.

CLUSTER_ID**Type**

int

Default value

0x0

Processor cluster ID value.

IMINLN**Type**

bool

Default value

0x1

Instruction cache minimum line size: false=32 bytes, true=64 bytes.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Initialize to BE8 endianness.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.DBGROMADDR**Type**

int

Default value

0x12000003

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

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

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

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

T32 exception enable. The default has exceptions including reset handled in A32 state.

cpuX.VINITHI**Type**

bool

Default value

0x0

Initialize with high vectors enabled.

cpuX.ase-present**Type**

bool

Default value

0x1

Set whether CT model has been built with NEON support.

cpuX.l1_icache-size**Type**

int

Default value

0x8000

Size of L1 I-cache.

cpuX.min_sync_level**Type**

int

Default value

0x0

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-ARM_HLT**Type**

int

Default value

0xf000

ARM HLT number for semihosting.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

ARM SVC number for semihosting.

cpuX.semihosting-Thumb_HLT**Type**

int

Default value

0x3c

Thumb HLT number for semihosting.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

Thumb SVC number for semihosting.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

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

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-hlt-enable**Type**

bool

Default value

0x0

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

Prefix semihosting output with target instance name.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

Enable coprocessor access and VFP at reset.

cpuX.vfp-present**Type**

bool

Default value

0x1

Set whether CT model has been built with VFP support.

l1_dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

Set whether L1 D-cache has stateful implementation.

l1_dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

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

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_ichache-read_access_latency**Type**

int

Default value

0x0

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_ichache-read_latency is set. This is in addition to the hit or miss latency, 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-read_latency**Type**

int

Default value

0x0

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

Set whether L1 I-cache has stateful implementation.

l2-data-slice**Type**

int

Default value

0x0

L2 data RAM slice.

l2-tag-slice**Type**

int

Default value

0x0

L2 tag RAM slice.

l2_cache-hit_latency

Type

int

Default value

0x0

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

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

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

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

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

Set L2 cache size in bytes.

`l2_cache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

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

Set whether L2 cache has stateful implementation.

l2_cache-write_access_latency

Type
int

Default value
0x0

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

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 **ARMCortexA32x1CT**

ARMCortexA32x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-232: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA32x1CT

This model has the following Iris instances:

Table 3-233: ARMCortexA32x1CT Iris instances

InstanceName	ComponentName
ARMCortexA32x1CT	Cluster_ARM_Cortex-A32
ARMCortexA32x1CT.AMU	PVBusLogger
ARMCortexA32x1CT.AMU.mapper	PVBusMapper
ARMCortexA32x1CT.DAP	PVBusLogger
ARMCortexA32x1CT.DAP.mapper	PVBusMapper
ARMCortexA32x1CT.DSU	DSU

InstanceName	ComponentName
ARMCortexA32x1CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA32x1CT.MMAP	PVBusLogger
ARMCortexA32x1CT.MMAP.mapper	PVBusMapper
ARMCortexA32x1CT.RAS	PVBusLogger
ARMCortexA32x1CT.RAS.mapper	PVBusMapper
ARMCortexA32x1CT.acp_mapper	PVBusMapper
ARMCortexA32x1CT.cpu0	ARM_Cortex-A32
ARMCortexA32x1CT.cpu0.S1TLB	TLB
ARMCortexA32x1CT.cpu0.S2TLB	TLB
ARMCortexA32x1CT.cpu0.UTLB	TLB
ARMCortexA32x1CT.cpu0.dtlb	TlbCadi
ARMCortexA32x1CT.cpu0.l1dcache	PVCache
ARMCortexA32x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA32x1CT.cpu0.l1licache	PVCache
ARMCortexA32x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA32x1CT.ext_bus	PVBusLogger
ARMCortexA32x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA32x1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA32x1CT.l2_cache	PVCache
ARMCortexA32x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA32x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-234: ARMCortexA32x1CT MTI instances

InstanceName	ComponentName
ARMCortexA32x1CT	ARMv8Cluster
ARMCortexA32x1CT.AMU	PVBusLogger
ARMCortexA32x1CT.AMU.mapper	PVBusMapper
ARMCortexA32x1CT.DAP	PVBusLogger
ARMCortexA32x1CT.DAP.mapper	PVBusMapper
ARMCortexA32x1CT.DSU	DSU
ARMCortexA32x1CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA32x1CT.MMAP	PVBusLogger
ARMCortexA32x1CT.MMAP.mapper	PVBusMapper
ARMCortexA32x1CT.RAS	PVBusLogger
ARMCortexA32x1CT.RAS.mapper	PVBusMapper
ARMCortexA32x1CT.acp_mapper	PVBusMapper
ARMCortexA32x1CT.cpu0	ARM_Cortex-A32
ARMCortexA32x1CT.cpu0.S1TLB	TLB
ARMCortexA32x1CT.cpu0.S2TLB	TLB
ARMCortexA32x1CT.cpu0.UTLB	TLB
ARMCortexA32x1CT.cpu0.l1dcache	PVCache
ARMCortexA32x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA32x1CT.cpu0.l1licache	PVCache
ARMCortexA32x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA32x1CT.ext_bus	PVBusLogger
ARMCortexA32x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA32x1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA32x1CT.l2_cache	PVCache
ARMCortexA32x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[6]	PVBusSlave

InstanceName	ComponentName
ARMCortexA32x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA32x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA32x1CT.l2_flusher	AsyncCacheFlushUnit

ARMCortexA32x1CT contains the following CADI targets:

- ARM_Cortex-A32
- Cluster_ARM_Cortex-A32
- PVCache
- TlbCadi

About ARMCortexA32x1CT

- 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.
- ECC and parity schemes are hardware-specific so are not supported.

Ports for ARMCortexA32x1CT

Table 3-235: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>broadcastcachemaint</code>	Signal	Slave	ACE defined pins.
<code>broadcastinner</code>	Signal	Slave	ACE defined pins.

Name	Protocol	Type	Description
broadcastouter	Signal	Slave	ACE 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 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.

Name	Protocol	Type	Description
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.
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.
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 ARMCortexA32x1CT

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTINNER

Type

bool

Default value

0x1

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CLUSTER_ID

Type

int

Default value

0x0

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x1

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x1

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

bus_type**Type**

int

Default value

0x0

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.CP15SDISABLE2**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers (FEAT_CP15SDISABLE2).

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.max_code_cache_mb**Type**

int

Default value

0x100

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`

Type

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency

Type
int

Default value
0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.13 ARMCortexA34x1CT

ARMCortexA34x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-236: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA34x1CT

This model has the following Iris instances:

Table 3-237: ARMCortexA34x1CT Iris instances

InstanceName	ComponentName
ARMCortexA34x1CT	Cluster_ARM_Cortex-A34
ARMCortexA34x1CT.AMU	PVBusLogger
ARMCortexA34x1CT.AMU.mapper	PVBusMapper
ARMCortexA34x1CT.DAP	PVBusLogger
ARMCortexA34x1CT.DAP.mapper	PVBusMapper
ARMCortexA34x1CT.DSU	DSU
ARMCortexA34x1CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA34x1CT.MMAP	PVBusLogger
ARMCortexA34x1CT.MMAP.mapper	PVBusMapper
ARMCortexA34x1CT.RAS	PVBusLogger
ARMCortexA34x1CT.RAS.mapper	PVBusMapper
ARMCortexA34x1CT.acp_mapper	PVBusMapper
ARMCortexA34x1CT.cpu0	ARM_Cortex-A34
ARMCortexA34x1CT.cpu0.UTLB	TLB
ARMCortexA34x1CT.cpu0.dtlb	TlbCadi
ARMCortexA34x1CT.cpu0.l1dcache	PVCache
ARMCortexA34x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA34x1CT.cpu0.l1licache	PVCache
ARMCortexA34x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA34x1CT.ext_bus	PVBusLogger
ARMCortexA34x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA34x1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA34x1CT.l2_cache	PVCache
ARMCortexA34x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[7]	PVBusSlave

InstanceName	ComponentName
ARMCortexA34x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA34x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-238: ARMCortexA34x1CT MTI instances

InstanceName	ComponentName
ARMCortexA34x1CT	ARMv8Cluster
ARMCortexA34x1CT.AMU	PVBusLogger
ARMCortexA34x1CT.AMU.mapper	PVBusMapper
ARMCortexA34x1CT.DAP	PVBusLogger
ARMCortexA34x1CT.DAP.mapper	PVBusMapper
ARMCortexA34x1CT.DSU	DSU
ARMCortexA34x1CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA34x1CT.MMAP	PVBusLogger
ARMCortexA34x1CT.MMAP.mapper	PVBusMapper
ARMCortexA34x1CT.RAS	PVBusLogger
ARMCortexA34x1CT.RAS.mapper	PVBusMapper
ARMCortexA34x1CT.acp_mapper	PVBusMapper
ARMCortexA34x1CT.cpu0	ARM_Cortex-A34
ARMCortexA34x1CT.cpu0.UTLB	TLB
ARMCortexA34x1CT.cpu0.l1dcache	PVCache
ARMCortexA34x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA34x1CT.cpu0.l1icache	PVCache
ARMCortexA34x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA34x1CT.ext_bus	PVBusLogger
ARMCortexA34x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA34x1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA34x1CT.l2_cache	PVCache
ARMCortexA34x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[2]	PVBusSlave

InstanceName	ComponentName
ARMCortexA34x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA34x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA34x1CT.l2_flusher	AsyncCacheFlushUnit

ARMCortexA34x1CT contains the following CADI targets:

- ARM_Cortex-A34
- Cluster_ARM_Cortex-A34
- PVCache
- TlbCadi

About ARMCortexA34x1CT

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.
- ECC and parity schemes are hardware-specific so are not supported.

Ports for ARMCortexA34x1CT

Table 3-239: 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.
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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 A34x1CT

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTINNER

Type

bool

Default value

0x1

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CLUSTER_ID

Type

int

Default value

0x0

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x1

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x1

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

bus_type**Type**

int

Default value

0x0

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.CP15SDISABLE2**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers (FEAT_CP15SDISABLE2).

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.max_code_cache_mb**Type**

int

Default value

0x100

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

`dcache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`**Type**

int

Default value

0x0

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

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

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

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled

Type

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l2cache-hit_latency

Type

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`l2cache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.14 ARM Cortex A35x1CT

ARM Cortex A35x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-240: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM Cortex A35x1CT

This model has the following Iris instances:

Table 3-241: ARM Cortex A35x1CT Iris instances

InstanceName	ComponentName
ARM Cortex A35x1CT	Cluster_ARM_Cortex-A35
ARM Cortex A35x1CT.AMU	PVBusLogger
ARM Cortex A35x1CT.AMU.mapper	PVBusMapper
ARM Cortex A35x1CT.DAP	PVBusLogger
ARM Cortex A35x1CT.DAP.mapper	PVBusMapper
ARM Cortex A35x1CT.DSU	DSU
ARM Cortex A35x1CT.DSU.mpam_busslave	PVBusSlave
ARM Cortex A35x1CT.MMAP	PVBusLogger
ARM Cortex A35x1CT.MMAP.mapper	PVBusMapper
ARM Cortex A35x1CT.RAS	PVBusLogger
ARM Cortex A35x1CT.RAS.mapper	PVBusMapper
ARM Cortex A35x1CT.acp_mapper	PVBusMapper
ARM Cortex A35x1CT.cpu0	ARM_Cortex-A35
ARM Cortex A35x1CT.cpu0.UTLB	TLB

InstanceName	ComponentName
ARMCortexA35x1CT.cpu0.dtlb	TlbCadi
ARMCortexA35x1CT.cpu0.l1dcache	PVCache
ARMCortexA35x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA35x1CT.cpu0.l1icache	PVCache
ARMCortexA35x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA35x1CT.ext_bus	PVBusLogger
ARMCortexA35x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA35x1CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder
ARMCortexA35x1CT.l2_cache	PVCache
ARMCortexA35x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA35x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-242: ARMCortexA35x1CT MTI instances

InstanceName	ComponentName
ARMCortexA35x1CT	ARMv8Cluster
ARMCortexA35x1CT.AMU	PVBusLogger
ARMCortexA35x1CT.AMU.mapper	PVBusMapper
ARMCortexA35x1CT.DAP	PVBusLogger
ARMCortexA35x1CT.DAP.mapper	PVBusMapper
ARMCortexA35x1CT.DSU	DSU
ARMCortexA35x1CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA35x1CT.MMAP	PVBusLogger

InstanceName	ComponentName
ARMCortexA35x1CT.MMAP.mapper	PVBusMapper
ARMCortexA35x1CT.RAS	PVBusLogger
ARMCortexA35x1CT.RAS.mapper	PVBusMapper
ARMCortexA35x1CT.acp_mapper	PVBusMapper
ARMCortexA35x1CT.cpu0	ARM_Cortex-A35
ARMCortexA35x1CT.cpu0.UTLB	TLB
ARMCortexA35x1CT.cpu0.l1dcache	PVCache
ARMCortexA35x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA35x1CT.cpu0.l1licache	PVCache
ARMCortexA35x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA35x1CT.ext_bus	PVBusLogger
ARMCortexA35x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA35x1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA35x1CT.l2_cache	PVCache
ARMCortexA35x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA35x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA35x1CT.l2_flusher	AsyncCacheFlushUnit

ARMCortexA35x1CT contains the following CADI targets:

- ARM_Cortex-A35
- Cluster_ARM_Cortex-A35
- PVCache
- TlbCadi

About ARMCortexA35x1CT

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.
- ECC and parity schemes are hardware-specific so are not supported.

Ports for ARMCortexA35x1CT

Table 3-243: 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 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 <code>clk_in</code> port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the <code>clrexmonreq</code> 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.
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.
dbgen[4]	Signal	Slave	External debug interface.
dbgnopwrdsn[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.

Name	Protocol	Type	Description
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
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 ARMCortexA35x1CT

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTINNER**Type**

bool

Default value

0x1

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CLUSTER_ID**Type**

int

Default value

0x0

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x1

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x1

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

bus_type**Type**

int

Default value

0x0

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

`cpi_mul`**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

`cpuX.AA64nAA32`**Type**

bool

Default value

0x1

Register width configuration at reset. 0, AArch32. 1, AArch64.

`cpuX.CFGEND`**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

`cpuX.CFGTE`**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

`cpuX.CP15SDISABLE`**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.CP15SDISABLE2**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers (FEAT_CP15SDISABLE2).

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.max_code_cache_mb**Type**

int

Default value

0x100

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`**Type**

int

Default value

0x0

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

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

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

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency

Type

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

`icache-state_modelled`**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

`l2cache-hit_latency`**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency

Type
int

Default value
0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.15 ARMCortexA53x1CT

ARMCortexA53x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-244: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM Cortex-A53x1CT

This model has the following Iris instances:

Table 3-245: ARM Cortex-A53x1CT Iris instances

InstanceName	ComponentName
ARM Cortex-A53x1CT	Cluster_ARM_Cortex-A53
ARM Cortex-A53x1CT.AMU	PVBusLogger
ARM Cortex-A53x1CT.AMU.mapper	PVBusMapper
ARM Cortex-A53x1CT.DAP	PVBusLogger
ARM Cortex-A53x1CT.DAP.mapper	PVBusMapper
ARM Cortex-A53x1CT.DSU	DSU
ARM Cortex-A53x1CT.DSU.mpam_busslave	PVBusSlave
ARM Cortex-A53x1CT.MMAP	PVBusLogger
ARM Cortex-A53x1CT.MMAP.mapper	PVBusMapper
ARM Cortex-A53x1CT.RAS	PVBusLogger
ARM Cortex-A53x1CT.RAS.mapper	PVBusMapper
ARM Cortex-A53x1CT.acp_mapper	PVBusMapper
ARM Cortex-A53x1CT.cpu0	ARM_Cortex-A53
ARM Cortex-A53x1CT.cpu0.UTLB	TLB
ARM Cortex-A53x1CT.cpu0.dtlb	TlbCadi
ARM Cortex-A53x1CT.cpu0.l1dcache	PVCache
ARM Cortex-A53x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM Cortex-A53x1CT.cpu0.l1licache	PVCache
ARM Cortex-A53x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARM Cortex-A53x1CT.ext_bus	PVBusLogger
ARM Cortex-A53x1CT.ext_bus.mapper	PVBusMapper
ARM Cortex-A53x1CT.gic_cpuif_decoder_cluster	GLICv3CPUInterfaceDecoder
ARM Cortex-A53x1CT.l2_cache	PVCache
ARM Cortex-A53x1CT.l2_cache.upstream[0]	PVBusSlave
ARM Cortex-A53x1CT.l2_cache.upstream[10]	PVBusSlave
ARM Cortex-A53x1CT.l2_cache.upstream[11]	PVBusSlave
ARM Cortex-A53x1CT.l2_cache.upstream[12]	PVBusSlave
ARM Cortex-A53x1CT.l2_cache.upstream[13]	PVBusSlave
ARM Cortex-A53x1CT.l2_cache.upstream[14]	PVBusSlave
ARM Cortex-A53x1CT.l2_cache.upstream[15]	PVBusSlave
ARM Cortex-A53x1CT.l2_cache.upstream[16]	PVBusSlave
ARM Cortex-A53x1CT.l2_cache.upstream[1]	PVBusSlave
ARM Cortex-A53x1CT.l2_cache.upstream[2]	PVBusSlave
ARM Cortex-A53x1CT.l2_cache.upstream[3]	PVBusSlave

InstanceName	ComponentName
ARMCortexA53x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA53x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-246: ARMCortexA53x1CT MTI instances

InstanceName	ComponentName
ARMCortexA53x1CT	ARMv8Cluster
ARMCortexA53x1CT.AMU	PVBusLogger
ARMCortexA53x1CT.AMU.mapper	PVBusMapper
ARMCortexA53x1CT.DAP	PVBusLogger
ARMCortexA53x1CT.DAP.mapper	PVBusMapper
ARMCortexA53x1CT.DSU	DSU
ARMCortexA53x1CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA53x1CT.MMAP	PVBusLogger
ARMCortexA53x1CT.MMAP.mapper	PVBusMapper
ARMCortexA53x1CT.RAS	PVBusLogger
ARMCortexA53x1CT.RAS.mapper	PVBusMapper
ARMCortexA53x1CT.acp_mapper	PVBusMapper
ARMCortexA53x1CT.cpu0	ARM_Cortex-A53
ARMCortexA53x1CT.cpu0.UTLB	TLB
ARMCortexA53x1CT.cpu0.l1dcache	PVCache
ARMCortexA53x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA53x1CT.cpu0.l1icache	PVCache
ARMCortexA53x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA53x1CT.ext_bus	PVBusLogger
ARMCortexA53x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA53x1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA53x1CT.l2_cache	PVCache
ARMCortexA53x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[14]	PVBusSlave

InstanceName	ComponentName
ARMCortexA53x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA53x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA53x1CT.l2_flusher	AsyncCacheFlushUnit

ARMCortexA53x1CT contains the following CADI targets:

- ARM_Cortex-A53
- Cluster_ARM_Cortex-A53
- PVCache
- TlbCadi

About ARMCortexA53x1CT

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_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.
 - 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.

Ports for ARMCortexA53x1CT

Table 3-247: Ports

Name	Protocol	Type	Description
<code>aa64naa32[4]</code>	Signal	Slave	Register width after reset.
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>broadcastcachemaint</code>	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
<code>broadcastinner</code>	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 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.
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 A53x1CT

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTINNER

Type

bool

Default value

0x1

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CLUSTER_ID

Type

int

Default value

0x0

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x1

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x1

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

bus_type**Type**

int

Default value

0x0

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.AA64nAA32**Type**

bool

Default value

0x1

Register width configuration at reset. 0, AArch32. 1, AArch64.

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.max_code_cache_mb**Type**

int

Default value

0x100

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

`dcache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`**Type**

int

Default value

0x0

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

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

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

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled

Type

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l2cache-hit_latency

Type

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`l2cache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

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

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

revision_number**Type**

int

Default value

0x0

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type

bool

Default value

0x0

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.16 ARMCortexA55CT

ARMCortexA55CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-248: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA55CT

This model has the following Iris instances:

Table 3-249: 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	TlbCadi
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
ARMCortexA55CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

This model has the following MTI trace components:

Table 3-250: ARMCortexA55CT MTI instances

InstanceName	ComponentName
ARMCortexA55CT	ARMv8Cluster
ARMCortexA55CT.AMU	PVBusLogger
ARMCortexA55CT.AMU.mapper	PVBusMapper
ARMCortexA55CT.DAP	PVBusLogger

InstanceName	ComponentName
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
ARMCortexA55CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

ARMCortexA55CT contains the following CADI targets:

- ARM_Cortex-A55
- Cluster_ARM_Cortex-A55
- PVCache
- TlbCadi

About ARMCortexA55CT

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 DBGNOFWRDWN 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](#).

Ports for ARMCortexA55CT

Table 3-251: 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.

Name	Protocol	Type	Description
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[8]	Signal	Slave	This signal is 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.
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.

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[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.
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 ARMCortexA55CT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

bus_type**Type**

int

Default value

0x0

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

`cpi_mul`**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

`cpuX.AA64nAA32`**Type**

bool

Default value

0x1

Register width configuration at reset. 0, AArch32. 1, AArch64.

`cpuX.CFGEND`**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

`cpuX.CFGTE`**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

`cpuX.CP15SDISABLE`**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`**Type**

int

Default value

0x0

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

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

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

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

If true, Accelerator Coherency Port is configured.

has_dot_product**Type**

int

Default value

0x2

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`

Type

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end**Type**

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.17 ARM CortexA55CT_CortexA75CT

ARM CortexA55CT_CortexA75CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-252: 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 23.

Iris and MTI instances for ARM CortexA55CT_CortexA75CT

This model has the following Iris instances:

Table 3-253: ARM CortexA55CT_CortexA75CT Iris instances

InstanceName	ComponentName
ARM CortexA55CT_CortexA75CT	Cluster_ARM_Cortex-A55_Cortex-A75
ARM CortexA55CT_CortexA75CT.AMU	PVBusLogger
ARM CortexA55CT_CortexA75CT.AMU.mapper	PVBusMapper
ARM CortexA55CT_CortexA75CT.DAP	PVBusLogger
ARM CortexA55CT_CortexA75CT.DAP.mapper	PVBusMapper
ARM CortexA55CT_CortexA75CT.DSU	DSU
ARM CortexA55CT_CortexA75CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM CortexA55CT_CortexA75CT.DSU.mpam_busslave	PVBusSlave
ARM CortexA55CT_CortexA75CT.DSU.shared_cache	PVCache
ARM CortexA55CT_CortexA75CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARM CortexA55CT_CortexA75CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARM CortexA55CT_CortexA75CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARM CortexA55CT_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.subcluster0	Subcluster_ARM_Cortex-A55
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0	ARM_Cortex-A55
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.dtlb	TlbCadi
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	TlbCadi

This model has the following MTI trace components:

Table 3-254: ARMCortexA55CT_CortexA75CT MTI instances

InstanceName	ComponentName
ARMCortexA55CT_CortexA75CT	ARMv8Cluster
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.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.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 contains the following CADI targets:

- ARM_Cortex-A55

- ARM_Cortex-A75
- Cluster_ARM_Cortex-A55_Cortex-A75
- PVCache
- Subcluster_ARM_Cortex-A55
- Subcluster_ARM_Cortex-A75
- TlbCadi

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](#).

Ports for ARMCortexA55CT_CortexA75CT

Table 3-255: 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[11:8] 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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

dcache-state_modelled**Type**

bool

Default value

0x0

Set whether D-cache has stateful implementation.

default_opmode**Type**

int

Default value

0x4

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

If true, Accelerator Coherency Port is configured.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end**Type**

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

subcluster0.NUM_CORES**Type**

int

Default value

0x1

Number of cores per cluster.

subcluster0.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster0.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster0.cpuX.AA64nAA32**Type**

bool

Default value

0x1

Register width configuration at reset. 0, AArch32. 1, AArch64.

subcluster0.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster0.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster0.cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

subcluster0.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster0.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster0.cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

subcluster0.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster0.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster0.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster0.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster0.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster0.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster0.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster0.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster0.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster0.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster0.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster0.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

`subcluster0.dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

subcluster0.icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster0.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.CCSIDR-L1D_override**Type**

int

Default value

0x0

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

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

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

Number of cores per cluster.

subcluster1.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster1.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster1.cpuX.AA64nAA32**Type**

bool

Default value

0x1

Register width configuration at reset. 0, AArch32. 1, AArch64.

subcluster1.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster1.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster1.cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

subcluster1.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster1.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster1.cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

subcluster1.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster1.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster1.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster1.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster1.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster1.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster1.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster1.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster1.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster1.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster1.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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.ptw_latency`

Type

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlb_latency

Type
int

Default value
0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlbi_stall_enabled

Type
bool

Default value
0x0

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

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 3-256: 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 23.

Iris and MTI instances for ARMCortexA55CT_CortexA76CT

This model has the following Iris instances:

Table 3-257: 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
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	GLCv3CPUInterfaceDecoder
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	TlbCadi
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l1icache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l1icache.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

InstanceName	ComponentName
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l1icache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l1icache.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	TlbCadi

This model has the following MTI trace components:

Table 3-258: ARMCortexA55CT_CortexA76CT MTI instances

InstanceName	ComponentName
ARMCortexA55CT_CortexA76CT	ARMv8Cluster
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
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.l1icache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l2cache	PVCache

InstanceName	ComponentName
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.l1icache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l1icache.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 contains the following CADI targets:

- ARM_Cortex-A55
- ARM_Cortex-A76
- Cluster_ARM_Cortex-A55_Cortex-A76
- PVCache
- Subcluster_ARM_Cortex-A55
- Subcluster_ARM_Cortex-A76
- TlbCadi

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](#)

Ports for ARMCortexA55CT_CortexA76CT

Table 3-259: Ports

Name	Protocol	Type	Description
<code>aa64naa32[11]</code>	Signal	Slave	Register width after reset.
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port
<code>AENDMP</code>	Value_64	Slave	DynamlQ port to obtain end address of valid peripheral address range (exclusive).
<code>ASTARTMP</code>	Value_64	Slave	DynamlQ port to obtain start address of valid peripheral address range (inclusive).
<code>broadcastatomic</code>	Signal	Slave	CHI defined pins.
<code>broadcastcachemaint</code>	Signal	Slave	ACE defined pins.
<code>broadcastouter</code>	Signal	Slave	ACE defined pins.
<code>broadcastpersist</code>	Signal	Slave	CHI defined pins.
<code>cfgend[11]</code>	Signal	Slave	This signal if for EE bit initialisation.
<code>cfgsdisable</code>	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
<code>cfgte[11]</code>	Signal	Slave	This signal provides default exception handling state.
<code>clk_in</code>	ClockSignal	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
<code>clusterid</code>	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
<code>clusterpmuirq</code>	Signal	Master	DynamlQ pmu irq
<code>CNTHPIRQ[11]</code>	Signal	Master	Timer signals to SOC.
<code>CNTHVIRQ[11]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPNSIRQ[11]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPSIRQ[11]</code>	Signal	Master	Timer signals to SOC.
<code>cntvalueb</code>	CounterInterface	Slave	Interface to SoC level counter module.
<code>CNTVIRQ[11]</code>	Signal	Master	Timer signals to SOC.
<code>commirq[11]</code>	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_CortexA76CT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

dcache-state_modelled**Type**

bool

Default value

0x0

Set whether D-cache has stateful implementation.

default_opmode**Type**

int

Default value

0x4

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

If true, Accelerator Coherency Port is configured.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end**Type**

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

subcluster0.NUM_CORES**Type**

int

Default value

0x1

Number of cores per cluster.

subcluster0.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster0.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster0.cpuX.AA64nAA32**Type**

bool

Default value

0x1

Register width configuration at reset. 0, AArch32. 1, AArch64.

subcluster0.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster0.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster0.cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

subcluster0.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster0.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster0.cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

subcluster0.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster0.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster0.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster0.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster0.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster0.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster0.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster0.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster0.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster0.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster0.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster0.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster0.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

`subcluster0.icache-hit_latency`**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

`subcluster0.ptw_latency`**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

`subcluster0.tlb_latency`**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

`subcluster0.tlbi_stall_enabled`**Type**

bool

Default value

0x0

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.NUM_CORES**Type**

int

Default value

0x1

Number of cores per cluster.

subcluster1.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster1.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster1.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster1.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster1.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster1.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster1.cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

subcluster1.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster1.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`subcluster1.cpuX.l2cache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster1.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster1.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster1.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster1.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster1.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster1.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster1.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster1.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

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

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

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

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

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

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

Synchronous reporting of device-nGnRE read external aborts.

subcluster1.ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

subcluster1.ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

subcluster1.ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

subcluster1.force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

Whether Statistical Based Profiling is implemented (FEAT_SPE).

subcluster1.icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

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 3-260: 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 23.

Iris and MTI instances for ARMCortexA55CT_CortexA78CT

This model has the following Iris instances:

Table 3-261: ARMCortexA55CT_CortexA78CT Iris instances

InstanceName	ComponentName
ARMCortexA55CT_CortexA78CT	Cluster_ARM_Cortex-A55_Cortex-A78
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	Subcluster_ARM_Cortex-A55

InstanceName	ComponentName
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0	ARM_Cortex-A55
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.dtlb	TlbCadi
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
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1.cpu1.dtlb	TlbCadi

This model has the following MTI trace components:

Table 3-262: ARMCortexA55CT_CortexA78CT MTI instances

InstanceName	ComponentName
ARMCortexA55CT_CortexA78CT	ARMv8Cluster
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

InstanceName	ComponentName
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.l1licache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l1licache.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
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l1licache.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

ARMCortexA55CT_CortexA78CT contains the following CADI targets:

- ARM_Cortex-A55
- ARM_Cortex-A78
- Cluster_ARM_Cortex-A55_Cortex-A78
- PVCache
- Subcluster_ARM_Cortex-A55
- Subcluster_ARM_Cortex-A78
- TlbCadi

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](#)

Ports for ARMCortexA55CT_CortexA78CT

Table 3-263: 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	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[11]	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[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.
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.
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

Name	Protocol	Type	Description
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.
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

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

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

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

dcache-state_modelled**Type**

bool

Default value

0x0

Set whether D-cache has stateful implementation.

default_opmode**Type**

int

Default value

0x4

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

Enable DynamlQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

If true, Accelerator Coherency Port is configured.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

`l3cache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end**Type**

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

subcluster0.NUM_CORES**Type**

int

Default value

0x1

Number of cores per cluster.

subcluster0.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster0.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster0.cpuX.AA64nAA32**Type**

bool

Default value

0x1

Register width configuration at reset. 0, AArch32. 1, AArch64.

subcluster0.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster0.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster0.cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

subcluster0.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster0.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster0.cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTL.R.V.

subcluster0.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster0.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster0.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster0.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster0.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster0.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster0.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster0.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster0.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster0.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster0.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster0.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster0.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

subcluster0.icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster0.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.NUM_CORES**Type**

int

Default value

0x1

Number of cores per cluster.

subcluster1.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster1.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster1.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster1.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster1.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster1.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster1.cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTL.R.V.

subcluster1.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster1.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster1.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster1.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster1.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster1.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster1.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster1.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster1.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster1.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster1.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster1.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

Synchronous reporting of device-nGnRE read external aborts.

subcluster1.ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

subcluster1.ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

subcluster1.ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

subcluster1.force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

subcluster1.icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster1.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlbi_stall_enabled

Type

bool

Default value

0x0

If true, tlbi 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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.20 ARMCortexA57x1CT

ARMCortexA57x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-264: IP revisions support

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM Cortex-A57x1CT

This model has the following Iris instances:

Table 3-265: ARM Cortex-A57x1CT Iris instances

InstanceName	ComponentName
ARM Cortex-A57x1CT	Cluster_ARM_Cortex-A57
ARM Cortex-A57x1CT.AMU	PVBusLogger
ARM Cortex-A57x1CT.AMU.mapper	PVBusMapper
ARM Cortex-A57x1CT.DAP	PVBusLogger
ARM Cortex-A57x1CT.DAP.mapper	PVBusMapper
ARM Cortex-A57x1CT.DSU	DSU
ARM Cortex-A57x1CT.DSU.mpam_busslave	PVBusSlave
ARM Cortex-A57x1CT.MMAP	PVBusLogger
ARM Cortex-A57x1CT.MMAP.mapper	PVBusMapper
ARM Cortex-A57x1CT.RAS	PVBusLogger
ARM Cortex-A57x1CT.RAS.mapper	PVBusMapper
ARM Cortex-A57x1CT.acp_mapper	PVBusMapper
ARM Cortex-A57x1CT.cpu0	ARM_Cortex-A57
ARM Cortex-A57x1CT.cpu0.UTLB	TLB
ARM Cortex-A57x1CT.cpu0.dtlb	TlbCadi
ARM Cortex-A57x1CT.cpu0.l1dcache	PVCache
ARM Cortex-A57x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM Cortex-A57x1CT.cpu0.l1icache	PVCache
ARM Cortex-A57x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARM Cortex-A57x1CT.ext_bus	PVBusLogger
ARM Cortex-A57x1CT.ext_bus.mapper	PVBusMapper
ARM Cortex-A57x1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARM Cortex-A57x1CT.l2_cache	PVCache
ARM Cortex-A57x1CT.l2_cache.upstream[0]	PVBusSlave
ARM Cortex-A57x1CT.l2_cache.upstream[10]	PVBusSlave
ARM Cortex-A57x1CT.l2_cache.upstream[11]	PVBusSlave
ARM Cortex-A57x1CT.l2_cache.upstream[12]	PVBusSlave
ARM Cortex-A57x1CT.l2_cache.upstream[13]	PVBusSlave
ARM Cortex-A57x1CT.l2_cache.upstream[14]	PVBusSlave
ARM Cortex-A57x1CT.l2_cache.upstream[15]	PVBusSlave
ARM Cortex-A57x1CT.l2_cache.upstream[16]	PVBusSlave
ARM Cortex-A57x1CT.l2_cache.upstream[1]	PVBusSlave
ARM Cortex-A57x1CT.l2_cache.upstream[2]	PVBusSlave
ARM Cortex-A57x1CT.l2_cache.upstream[3]	PVBusSlave
ARM Cortex-A57x1CT.l2_cache.upstream[4]	PVBusSlave

InstanceName	ComponentName
ARMCortexA57x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA57x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-266: ARMCortexA57x1CT MTI instances

InstanceName	ComponentName
ARMCortexA57x1CT	ARMv8Cluster
ARMCortexA57x1CT.AMU	PVBusLogger
ARMCortexA57x1CT.AMU.mapper	PVBusMapper
ARMCortexA57x1CT.DAP	PVBusLogger
ARMCortexA57x1CT.DAP.mapper	PVBusMapper
ARMCortexA57x1CT.DSU	DSU
ARMCortexA57x1CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA57x1CT.MMAP	PVBusLogger
ARMCortexA57x1CT.MMAP.mapper	PVBusMapper
ARMCortexA57x1CT.RAS	PVBusLogger
ARMCortexA57x1CT.RAS.mapper	PVBusMapper
ARMCortexA57x1CT.acp_mapper	PVBusMapper
ARMCortexA57x1CT.cpu0	ARM_Cortex-A57
ARMCortexA57x1CT.cpu0.UTLB	TLB
ARMCortexA57x1CT.cpu0.l1dcache	PVCache
ARMCortexA57x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA57x1CT.cpu0.l1licache	PVCache
ARMCortexA57x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA57x1CT.ext_bus	PVBusLogger
ARMCortexA57x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA57x1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA57x1CT.l2_cache	PVCache
ARMCortexA57x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[15]	PVBusSlave

InstanceName	ComponentName
ARMCortexA57x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA57x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA57x1CT.l2_flusher	AsyncCacheFlushUnit

ARMCortexA57x1CT contains the following CADI targets:

- ARM_Cortex-A57
- Cluster_ARM_Cortex-A57
- PVCache
- TlbCadi

About ARMCortexA57x1CT

The following components also exist:

- ARMCortexA57x2CT.
- ARMCortexA57x3CT.
- ARMCortexA57x4CT.

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.
 - The DBGPWRDUP, DBGPWRUPREQ, DBGNOPWRDWN, and DBGRSTREQ debug power management signals.
 - ECC and parity schemes are hardware-specific so are not supported.

Ports for ARMCortexA57x1CT

Table 3-267: Ports

Name	Protocol	Type	Description
aa64naa32[4]	Signal	Slave	Register width after reset.
acp_s	PVBUS	Slave	AXI ACP slave port.

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.
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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 A57x1CT

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTINNER

Type

bool

Default value

0x1

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CLUSTER_ID

Type

int

Default value

0x0

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x1

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x1

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

bus_type**Type**

int

Default value

0x0

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.AA64nAA32**Type**

bool

Default value

0x1

Register width configuration at reset. 0, AArch32. 1, AArch64.

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.max_code_cache_mb**Type**

int

Default value

0x100

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`**Type**

int

Default value

0x0

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

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

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

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

Set whether I-cache has stateful implementation.

`l2cache-hit_latency`**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency

Type
int

Default value
0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

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 3-268: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA65AECT

This model has the following Iris instances:

Table 3-269: 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	TlbCadi
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.dtlb	TlbCadi
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

InstanceName	ComponentName
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	GLCv3CPUInterfaceDecoder

This model has the following MTI trace components:

Table 3-270: ARMCortexA65AECT MTI instances

InstanceName	ComponentName
ARMCortexA65AECT	ARMv8Cluster
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.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

InstanceName	ComponentName
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	GlCv3CPUInterfaceDecoder

ARMCortexA65AECT contains the following CADI targets:

- ARM_Cortex-A65AE
- Cluster_ARM_Cortex-A65AE
- PVCache
- TlbCadi

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.

Ports for ARMCortexA65AECT

Table 3-271: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	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[16]	Signal	Slave	This signal if 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 core executes instructions.
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[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.

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 ARMCortexA65AECT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

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

Number of cores per cluster.

cluster_patch_level

Type

int

Default value

0x0

Cosmetic change to patch number in CLUSTERIDR. Corresponds to the Y in rXpY.

cluster_revision_number

Type

int

Default value

0x0

Cosmetic change to revision number in CLUSTERIDR, Corresponds to the X in rXpY.

`cpi_div`**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

`cpi_mul`**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

`cpuX.CRYPTODISABLE`**Type**

bool

Default value

0x0

Disable cryptographic features.

`cpuX.enable_single_thread_at_reset`**Type**

bool

Default value

0x0

Enable single thread after reset and keep other thread in reset.

`cpuX.enable_trace_special_hlt_imm16`**Type**

bool

Default value

0x0

Enable usage of parameter `trace_special_hlt_imm16`.

`cpuX.l2cache-hit_latency`

Type

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`cpuX.l2cache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.threadY.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.threadY.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.threadY.MPIDR-override**Type**

int

Default value

0x0

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

Value of RVBAR_ELx register.

cpuX.threadY.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTL.R.V.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_lock_step**Type**

bool

Default value

0x0

Whether the core is configured in Dual Core Lock Step mode (FEAT_DCLS).

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

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

If true, Accelerator Coherency Port is configured.

has_dot_product**Type**

int

Default value

0x2

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`

Type

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

`periph_address_end`**Type**

int

Default value

0x0

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

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 3-272: 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 23.

Iris and MTI instances for ARMCortexA65AECT_CortexA76AECT

This model has the following Iris instances:

Table 3-273: 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.13_flusher	AsyncCacheFlushUnit

InstanceName	ComponentName
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
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.subcluster0	Subcluster_ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.dtlb	TlbCadi
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.dtlb	TlbCadi
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0	ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l1dcache	PVCache

InstanceName	ComponentName
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	Subcluster_ARM_Cortex-A76AE
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.l1icache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l1icache.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.l1icache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l1icache.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	TlbCadi
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu3.dtlb	TlbCadi

This model has the following MTI trace components:

Table 3-274: ARMCortexA65AECT_CortexA76AECT MTI instances

InstanceName	ComponentName
ARMCortexA65AECT_CortexA76AECT	ARMv8Cluster
ARMCortexA65AECT_CortexA76AECT.AMU	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.AMU.mapper	PVBusMapper

InstanceName	ComponentName
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
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.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

InstanceName	ComponentName
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
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l1icache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l1icache.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.l1icache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l1icache.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 contains the following CADI targets:

- ARM_Cortex-A65AE
- ARM_Cortex-A76AE
- Cluster_ARM_Cortex-A65AE_Cortex-A76AE
- PVCache
- Subcluster_ARM_Cortex-A65AE
- Subcluster_ARM_Cortex-A76AE
- TlbCadi

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](#)

Ports for ARMCortexA65AECT_CortexA76AECT

Table 3-275: Ports

Name	Protocol	Type	Description
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[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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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[10]	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.
dbgprupreq[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.
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.

Name	Protocol	Type	Description
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbush_m0	PVBus	Master	The core will generate bus requests on this port.
pvbush_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 ARM CortexA65AECT_CortexA76AECT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Set whether D-cache has stateful implementation.

default_opmode

Type

int

Default value

0x4

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

Enable DynamIQ diagnostic messages.

enable_lock_step

Type

bool

Default value

0x0

Whether the core is configured in Dual Core Lock Step mode (FEAT_DCLS).

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

If true, Accelerator Coherency Port is configured.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency

Type

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end**Type**

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

subcluster0.NUM_CORES**Type**

int

Default value

0x2

Number of cores per cluster.

subcluster0.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster0.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster0.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster0.cpuX.enable_single_thread_at_reset**Type**

bool

Default value

0x0

Enable single thread after reset and keep other thread in reset.

subcluster0.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster0.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster0.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster0.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-T32_HLT

Type

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-Thumb_SVC

Type

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-cmd_line

Type

string

Default value

""

Command line available to semihosting calls.

subcluster0.cpuX.semihosting-cwd

Type

string

Default value

""

Base directory for semihosting file access.

subcluster0.cpuX.semihosting-enable

Type

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster0.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster0.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster0.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster0.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster0.cpuX.threadY.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster0.cpuX.threadY.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster0.cpuX.threadY.MPIDR-override**Type**

int

Default value

0x0

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

Value of RVBAR_ELx register.

subcluster0.cpuX.threadY.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

subcluster0.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster0.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster0.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

subcluster0.icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

`subcluster0.ptw_latency`

Type

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

`subcluster0.reported_patch_level`

Type

int

Default value

0xffffffffffffffff

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

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.NUM_CORES**Type**

int

Default value

0x2

Number of cores per cluster.

subcluster1.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster1.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster1.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster1.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster1.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster1.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster1.cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

subcluster1.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster1.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster1.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster1.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster1.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster1.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster1.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster1.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster1.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster1.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster1.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

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

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

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

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

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

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

Synchronous reporting of device-nGnRE read external aborts.

subcluster1.ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

subcluster1.ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

subcluster1.ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

subcluster1.force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

subcluster1.has_statistical_profiling**Type**

bool

Default value

0x1

Whether Statistical Based Profiling is implemented (FEAT_SPE).

subcluster1.icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.reported_patch_level**Type**

int

Default value`0xffffffffffffffff`

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`

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`

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

`subcluster1.tlbi_stall_enabled`**Type**`bool`**Default value**`0x0`

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`

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.23 ARM CortexA65CT

ARM CortexA65CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-276: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM CortexA65CT

This model has the following Iris instances:

Table 3-277: ARM CortexA65CT Iris instances

InstanceName	ComponentName
ARM CortexA65CT	Cluster_ARM_Cortex-A65
ARM CortexA65CT.AMU	PVBusLogger
ARM CortexA65CT.AMU.mapper	PVBusMapper
ARM CortexA65CT.DAP	PVBusLogger
ARM CortexA65CT.DAP.mapper	PVBusMapper
ARM CortexA65CT.DSU	DSU
ARM CortexA65CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM CortexA65CT.DSU.mpam_busslave	PVBusSlave
ARM CortexA65CT.DSU.shared_cache	PVCache
ARM CortexA65CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARM CortexA65CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARM CortexA65CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARM CortexA65CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARM CortexA65CT.MMAP	PVBusLogger
ARM CortexA65CT.MMAP.mapper	PVBusMapper
ARM CortexA65CT.RAS	PVBusLogger
ARM CortexA65CT.RAS.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA65CT.cpu0.dtlb	TlbCadi
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

This model has the following MTI trace components:

Table 3-278: ARMCortexA65CT MTI instances

InstanceName	ComponentName
ARMCortexA65CT	ARMv8Cluster
ARMCortexA65CT.AMU	PVBusLogger
ARMCortexA65CT.AMU.mapper	PVBusMapper
ARMCortexA65CT.DAP	PVBusLogger
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.thread0	ARM_Cortex-A65
ARMCortexA65CT.cpu0.thread0.UTLB	TLB
ARMCortexA65CT.cpu0.thread0.l1dcache	PVCache

InstanceName	ComponentName
ARMCortexA65CT.cpu0.thread0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65CT.cpu0.thread0.l1icache	PVCache
ARMCortexA65CT.cpu0.thread0.l1icache.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 contains the following CADI targets:

- ARM_Cortex-A65
- Cluster_ARM_Cortex-A65
- PVCache
- TlbCadi

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 `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](#).

Ports for ARMCortexA65CT

Table 3-279: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>AENDMP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>ASTARTMP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>broadcastatomic</code>	Signal	Slave	CHI defined pins.
<code>broadcastcachemaint</code>	Signal	Slave	ACE defined pins.
<code>broadcastouter</code>	Signal	Slave	ACE defined pins.
<code>broadcastpersist</code>	Signal	Slave	CHI defined pins.
<code>cfgend[16]</code>	Signal	Slave	This signal is for EE bit initialisation.
<code>cfgte[16]</code>	Signal	Slave	This signal provides default exception handling state.
<code>clk_in</code>	ClockSignal	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
<code>clusterid</code>	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
<code>clusterpmuirq</code>	Signal	Master	DynamlQ pmu irq
<code>CNTHPIRQ[16]</code>	Signal	Master	Timer signals to SOC.
<code>CNTHVIRQ[16]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPNSIRQ[16]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPSIRQ[16]</code>	Signal	Master	Timer signals to SOC.
<code>cntvalueb</code>	CounterInterface	Slave	Interface to SoC level counter module.

Name	Protocol	Type	Description
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.
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

Name	Protocol	Type	Description
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

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

cluster_patch_level**Type**

int

Default value

0x0

Cosmetic change to patch number in CLUSTERIDR. Corresponds to the Y in rXpY.

cluster_revision_number**Type**

int

Default value

0x0

Cosmetic change to revision number in CLUSTERIDR, Corresponds to the X in rXpY.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.enable_single_thread_at_reset**Type**

bool

Default value

0x0

Enable single thread after reset and keep other thread in reset.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.threadY.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.threadY.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.threadY.MPIDR-override**Type**

int

Default value

0x0

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

Value of RVBAR_ELx register.

cpuX.threadY.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

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

If true, Accelerator Coherency Port is configured.

has_dot_product**Type**

int

Default value

0x2

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

`icache-state_modelled`

Type

bool

Default value

0x0

Set whether I-cache has stateful implementation.

`l3cache-hit_latency`

Type

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end

Type

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start

Type

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency

Type

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency

Type

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.24 ARMCortexA72x1CT

ARMCortexA72x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-280: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA72x1CT

This model has the following Iris instances:

Table 3-281: ARMCortexA72x1CT Iris instances

InstanceName	ComponentName
ARMCortexA72x1CT	Cluster_ARM_Cortex-A72
ARMCortexA72x1CT.AMU	PVBUSLogger
ARMCortexA72x1CT.AMU.mapper	PVBUSMapper
ARMCortexA72x1CT.DAP	PVBUSLogger
ARMCortexA72x1CT.DAP.mapper	PVBUSMapper
ARMCortexA72x1CT.DSU	DSU
ARMCortexA72x1CT.DSU.mpam_busslave	PVBUSSlave
ARMCortexA72x1CT.MMAP	PVBUSLogger

InstanceName	ComponentName
ARMCortexA72x1CT.MMAP.mapper	PVBusMapper
ARMCortexA72x1CT.RAS	PVBusLogger
ARMCortexA72x1CT.RAS.mapper	PVBusMapper
ARMCortexA72x1CT.acp_mapper	PVBusMapper
ARMCortexA72x1CT.cpu0	ARM_Cortex-A72
ARMCortexA72x1CT.cpu0.UTLB	TLB
ARMCortexA72x1CT.cpu0.dtlb	TlbCadi
ARMCortexA72x1CT.cpu0.l1dcache	PVCache
ARMCortexA72x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA72x1CT.cpu0.l1licache	PVCache
ARMCortexA72x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA72x1CT.ext_bus	PVBusLogger
ARMCortexA72x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA72x1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA72x1CT.l2_cache	PVCache
ARMCortexA72x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA72x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-282: ARMCortexA72x1CT MTI instances

InstanceName	ComponentName
ARMCortexA72x1CT	ARMv8Cluster
ARMCortexA72x1CT.AMU	PVBusLogger

InstanceName	ComponentName
ARMCortexA72x1CT.AMU.mapper	PVBusMapper
ARMCortexA72x1CT.DAP	PVBusLogger
ARMCortexA72x1CT.DAP.mapper	PVBusMapper
ARMCortexA72x1CT.DSU	DSU
ARMCortexA72x1CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA72x1CT.MMAP	PVBusLogger
ARMCortexA72x1CT.MMAP.mapper	PVBusMapper
ARMCortexA72x1CT.RAS	PVBusLogger
ARMCortexA72x1CT.RAS.mapper	PVBusMapper
ARMCortexA72x1CT.acp_mapper	PVBusMapper
ARMCortexA72x1CT.cpu0	ARM_Cortex-A72
ARMCortexA72x1CT.cpu0.UTLB	TLB
ARMCortexA72x1CT.cpu0.l1dcache	PVCache
ARMCortexA72x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA72x1CT.cpu0.l1licache	PVCache
ARMCortexA72x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA72x1CT.ext_bus	PVBusLogger
ARMCortexA72x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA72x1CT.gic_cpuif_decoder_cluster	GLICv3CPUInterfaceDecoder
ARMCortexA72x1CT.l2_cache	PVCache
ARMCortexA72x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA72x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA72x1CT.l2_flusher	AsyncCacheFlushUnit

ARMCortexA72x1CT contains the following CADI targets:

- ARM_Cortex-A72
- Cluster_ARM_Cortex-A72
- PVCache
- TlbCadi

About ARMCortexA72x1CT

- The following components also exist:
 - ARMCortexA72x2CT.
 - ARMCortexA72x3CT.
 - ARMCortexA72x4CT.

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 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](#).

Ports for ARMCortexA72x1CT

Table 3-283: 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 <code>clk_in</code> port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the <code>clrexmonreq</code> signal

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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 ARMCortexA72x1CT

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTINNER**Type**

bool

Default value

0x1

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CLUSTER_ID**Type**

int

Default value

0x0

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x1

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x1

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

bus_type**Type**

int

Default value

0x0

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.AA64nAA32**Type**

bool

Default value

0x1

Register width configuration at reset. 0, AArch32. 1, AArch64.

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.max_code_cache_mb**Type**

int

Default value

0x100

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`

Type

int

Default value

0x0

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

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

Set whether I-cache has stateful implementation.

l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`l2cache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.25 ARMCortexA73x1CT

ARMCortexA73x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-284: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA73x1CT

This model has the following Iris instances:

Table 3-285: ARMCortexA73x1CT Iris instances

InstanceName	ComponentName
ARMCortexA73x1CT	Cluster_ARM_Cortex-A73
ARMCortexA73x1CT.AMU	PVBusLogger
ARMCortexA73x1CT.AMU.mapper	PVBusMapper
ARMCortexA73x1CT.DAP	PVBusLogger
ARMCortexA73x1CT.DAP.mapper	PVBusMapper
ARMCortexA73x1CT.DSU	DSU
ARMCortexA73x1CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA73x1CT.MMAP	PVBusLogger
ARMCortexA73x1CT.MMAP.mapper	PVBusMapper
ARMCortexA73x1CT.RAS	PVBusLogger
ARMCortexA73x1CT.RAS.mapper	PVBusMapper
ARMCortexA73x1CT.acp_mapper	PVBusMapper
ARMCortexA73x1CT.cpu0	ARM_Cortex-A73
ARMCortexA73x1CT.cpu0.UTLB	TLB
ARMCortexA73x1CT.cpu0.dtlb	TlbCadi
ARMCortexA73x1CT.cpu0.l1dcache	PVCache
ARMCortexA73x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexA73x1CT.cpu0.l1licache	PVCache
ARMCortexA73x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA73x1CT.ext_bus	PVBusLogger
ARMCortexA73x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA73x1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA73x1CT.l2_cache	PVCache
ARMCortexA73x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA73x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-286: ARMCortexA73x1CT MTI instances

InstanceName	ComponentName
ARMCortexA73x1CT	ARMv8Cluster
ARMCortexA73x1CT.AMU	PVBusLogger
ARMCortexA73x1CT.AMU.mapper	PVBusMapper
ARMCortexA73x1CT.DAP	PVBusLogger
ARMCortexA73x1CT.DAP.mapper	PVBusMapper
ARMCortexA73x1CT.DSU	DSU
ARMCortexA73x1CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA73x1CT.MMAP	PVBusLogger
ARMCortexA73x1CT.MMAP.mapper	PVBusMapper
ARMCortexA73x1CT.RAS	PVBusLogger
ARMCortexA73x1CT.RAS.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA73x1CT.acp_mapper	PVBusMapper
ARMCortexA73x1CT.cpu0	ARM_Cortex-A73
ARMCortexA73x1CT.cpu0.UTLB	TLB
ARMCortexA73x1CT.cpu0.l1dcache	PVCache
ARMCortexA73x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA73x1CT.cpu0.l1licache	PVCache
ARMCortexA73x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA73x1CT.ext_bus	PVBusLogger
ARMCortexA73x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA73x1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA73x1CT.l2_cache	PVCache
ARMCortexA73x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA73x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA73x1CT.l2_flusher	AsyncCacheFlushUnit

ARMCortexA73x1CT contains the following CADI targets:

- ARM_Cortex-A73
- Cluster_ARM_Cortex-A73
- PVCache
- TlbCadi

About ARMCortexA73x1CT

- The following components also exist:
 - ARMCortexA73x2CT.

- ARMCortexA73x3CT.
- ARMCortexA73x4CT.
- The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-3).
- 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](#).
- 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.
- 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`.
- ECC support is hardware-specific so is not modeled.

Ports for ARMCortexA73x1CT

Table 3-287: Ports

Name	Protocol	Type	Description
<code>aa64naa32[4]</code>	Signal	Slave	Register width after reset.
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>broadcastcachemaint</code>	Signal	Slave	ACE defined pins.
<code>broadcastinner</code>	Signal	Slave	ACE defined pins.
<code>broadcastouter</code>	Signal	Slave	ACE defined pins.
<code>cfgend[4]</code>	Signal	Slave	This signal if for EE bit initialisation.
<code>cfgsdisable</code>	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
<code>cfgte[4]</code>	Signal	Slave	This signal provides default exception handling state.
<code>clk_in</code>	ClockSignal	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the core executes instructions.
<code>clrexmonack</code>	Signal	Master	Acknowledge handshake signal for the <code>clrexmonreq</code> signal
<code>clrexmonreq</code>	Signal	Slave	Signals the clearing of an external global exclusive monitor
<code>clusterid</code>	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
<code>CNTHPIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPNSIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPSIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>cntvalueb</code>	CounterInterface	Slave	Interface to SoC level counter module.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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 indicates all cores and L2 are idle 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 ARMCortexA73x1CT

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTINNER

Type

bool

Default value

0x1

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CCSIDR-L1D_override**Type**

int

Default value

0x0

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

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

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x1

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x1

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

`cpi_div`**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

`cpi_mul`**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

`cpuX.AA64nAA32`**Type**

bool

Default value

0x1

Register width configuration at reset. 0, AArch32. 1, AArch64.

`cpuX.CFGEND`**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

`cpuX.CFGTE`**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.SMPEN**Type**

bool

Default value

0x0

Enable broadcast messages necessary for correct SMP operation at reset.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTL.R.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.max_code_cache_mb**Type**

int

Default value

0x100

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

Set whether I-cache has stateful implementation.

l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

revision_number**Type**

int

Default value

0x0

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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 3-288: IP revisions support

Revision	Quality level
r3p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA75CT

This model has the following Iris instances:

Table 3-289: ARMCortexA75CT Iris instances

InstanceName	ComponentName
ARMCortexA75CT	Cluster_ARM_Cortex-A75
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
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	TlbCadi
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

This model has the following MTI trace components:

Table 3-290: ARMCortexA75CT MTI instances

InstanceName	ComponentName
ARMCortexA75CT	ARMv8Cluster
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
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

ARMCortexA75CT contains the following CADI targets:

- ARM_Cortex-A75
- Cluster_ARM_Cortex-A75
- PVCache
- TlbCadi

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 DBGWRUP and DBGWRSTREQ are not implemented, but DBGWRUPREQ and DBGWRNOPWRDWN 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 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](#).

Ports for ARMCortexA75CT

Table 3-291: 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).

Name	Protocol	Type	Description
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 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.

Name	Protocol	Type	Description
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
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.

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 Cortex A75CT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

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

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

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

cpi_div

Type

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul

Type

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.AA64nAA32

Type

bool

Default value

0x1

Register width configuration at reset. 0, AArch32. 1, AArch64.

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`

Type

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

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

If true, Accelerator Coherency Port is configured.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`

Type

int

Default value

0x0

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

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

Set whether I-cache has stateful implementation.

`l3cache-hit_latency`

Type

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end

Type

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start

Type

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency

Type

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency

Type

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

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 3-292: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA76AECT

This model has the following Iris instances:

Table 3-293: ARMCortexA76AECT Iris instances

InstanceName	ComponentName
ARMCortexA76AECT	Cluster_ARM_Cortex-A76AE
ARMCortexA76AECT.AMU	PVBusLogger
ARMCortexA76AECT.AMU.mapper	PVBusMapper
ARMCortexA76AECT.DAP	PVBusLogger
ARMCortexA76AECT.DAP.mapper	PVBusMapper
ARMCortexA76AECT.DSU	DSU
ARMCortexA76AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA76AECT.DSU.mpam_busslave	PVBusSlave

InstanceName	ComponentName
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.dtlb	TlbCadi
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.dtlb	TlbCadi
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

This model has the following MTI trace components:

Table 3-294: ARMCortexA76AECT MTI instances

InstanceName	ComponentName
ARMCortexA76AECT	ARMv8Cluster

InstanceName	ComponentName
ARMCortexA76AECT.AMU	PVBusLogger
ARMCortexA76AECT.AMU.mapper	PVBusMapper
ARMCortexA76AECT.DAP	PVBusLogger
ARMCortexA76AECT.DAP.mapper	PVBusMapper
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

ARMCortexA76AECT contains the following CADI targets:

- ARM_Cortex-A76AE
- Cluster_ARM_Cortex-A76AE
- PVCache
- TlbCadi

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 DBGRSTREQ 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 `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](#).

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.

Ports for ARMCortexA76AECT

Table 3-295: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>AENDMP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>ASTARTMP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>broadcastatomic</code>	Signal	Slave	CHI defined pins.
<code>broadcastcachemaint</code>	Signal	Slave	ACE defined pins.
<code>broadcastouter</code>	Signal	Slave	ACE defined pins.
<code>broadcastpersist</code>	Signal	Slave	CHI defined pins.
<code>cfgend[4]</code>	Signal	Slave	This signal if for EE bit initialisation.
<code>cfgte[4]</code>	Signal	Slave	This signal provides default exception handling state.
<code>clk_in</code>	ClockSignal	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
<code>clusterid</code>	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
<code>clusterpmuirq</code>	Signal	Master	DynamlQ pmu irq
<code>CNTHPIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>CNTHVIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPNSIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPSIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>cntvalueb</code>	CounterInterface	Slave	Interface to SoC level counter module.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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 ARMCortexA76AECT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

`enable_lock_step`**Type**

bool

Default value

0x0

Whether the core is configured in Dual Core Lock Step mode (FEAT_DCLS).

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

If true, Accelerator Coherency Port is configured.

has_dot_product**Type**

int

Default value

0x2

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

has_statistical_profiling**Type**

bool

Default value

0x1

Whether Statistical Based Profiling is implemented (FEAT_SPE).

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

Set whether I-cache has stateful implementation.

`l3cache-hit_latency`

Type

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end**Type**

int

Default value

0x0

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

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 3-296: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA76CT

This model has the following Iris instances:

Table 3-297: ARMCortexA76CT Iris instances

InstanceName	ComponentName
ARMCortexA76CT	Cluster_ARM_Cortex-A76
ARMCortexA76CT.AMU	PVBusLogger
ARMCortexA76CT.AMU.mapper	PVBusMapper
ARMCortexA76CT.DAP	PVBusLogger
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.dtlb	TlbCadi
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

This model has the following MTI trace components:

Table 3-298: ARMCortexA76CT MTI instances

InstanceName	ComponentName
ARMCortexA76CT	ARMv8Cluster
ARMCortexA76CT.AMU	PVBusLogger
ARMCortexA76CT.AMU.mapper	PVBusMapper
ARMCortexA76CT.DAP	PVBusLogger
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.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

ARMCortexA76CT contains the following CADI targets:

- ARM_Cortex-A76
- Cluster_ARM_Cortex-A76
- PVCache
- TlbCadi

About ARMCortexA76CT

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.

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](#).

Ports for ARMCortexA76CT

Table 3-299: 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).

Name	Protocol	Type	Description
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 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	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.

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 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).

Name	Protocol	Type	Description
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM Cortex A76CT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

`cpi_mul`**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

`cpuX.CFGEND`**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

`cpuX.CFGTE`**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

`cpuX.CRYPTODISABLE`**Type**

bool

Default value

0x0

Disable cryptographic features.

`cpuX.RVBARADDR`**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`

Type

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

If true, Accelerator Coherency Port is configured.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

has_statistical_profiling**Type**

bool

Default value

0x1

Whether Statistical Based Profiling is implemented (FEAT_SPE).

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

Set whether I-cache has stateful implementation.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency

Type

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end**Type**

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

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

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 3-300: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA77CT

This model has the following Iris instances:

Table 3-301: ARMCortexA77CT Iris instances

InstanceName	ComponentName
ARMCortexA77CT	Cluster_ARM_Cortex-A77
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
ARMCortexA77CT.RAS.mapper	PVBusMapper
ARMCortexA77CT.cpu0	ARM_Cortex-A77
ARMCortexA77CT.cpu0.UTLB	TLB
ARMCortexA77CT.cpu0.dtlb	TlbCadi
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

This model has the following MTI trace components:

Table 3-302: ARMCortexA77CT MTI instances

InstanceName	ComponentName
ARMCortexA77CT	ARMv8Cluster
ARMCortexA77CT.AMU	PVBusLogger
ARMCortexA77CT.AMU.mapper	PVBusMapper
ARMCortexA77CT.DAP	PVBusLogger

InstanceName	ComponentName
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
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

ARMCortexA77CT contains the following CADI targets:

- ARM_Cortex-A77
- Cluster_ARM_Cortex-A77
- PVCache
- TlbCadi

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 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](#).

Ports for ARM Cortex-A77CT

Table 3-303: 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.

Name	Protocol	Type	Description
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 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.
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.

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	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 A77CT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

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

Number of cores per cluster.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter `trace_special_hlt_imm16`.

`cpuX.l2cache-hit_latency`

Type

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`cpuX.l2cache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

`dcache-read_access_latency`

Type

int

Default value

0x0

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

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

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations

Type

bool

Default value

0x1

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

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_write_is_sync

Type

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_so_read_is_sync

Type

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_write_is_sync

Type

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

If true, Accelerator Coherency Port is configured.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

has_statistical_profiling**Type**

bool

Default value

0x1

Whether Statistical Based Profiling is implemented (FEAT_SPE).

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

Set whether I-cache has stateful implementation.

`l3cache-hit_latency`**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

`periph_address_end`**Type**

int

Default value

0x0

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.30 ARM CortexA78AECT

ARM CortexA78AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-304: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM CortexA78AECT

This model has the following Iris instances:

Table 3-305: ARM CortexA78AECT Iris instances

InstanceName	ComponentName
ARM CortexA78AECT	Cluster_ARM_Cortex-A78AE
ARM CortexA78AECT.AMU	PVBusLogger
ARM CortexA78AECT.AMU.mapper	PVBusMapper
ARM CortexA78AECT.DAP	PVBusLogger
ARM CortexA78AECT.DAP.mapper	PVBusMapper
ARM CortexA78AECT.DSU	DSU
ARM CortexA78AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM CortexA78AECT.DSU.mpam_busslave	PVBusSlave
ARM CortexA78AECT.DSU.shared_cache	PVCache
ARM CortexA78AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARM CortexA78AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARM CortexA78AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARM CortexA78AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARM CortexA78AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARM CortexA78AECT.DSU.shared_cache.upstream[5]	PVBusSlave
ARM CortexA78AECT.MMAP	PVBusLogger
ARM CortexA78AECT.MMAP.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA78AECT.RAS	PVBusLogger
ARMCortexA78AECT.RAS.mapper	PVBusMapper
ARMCortexA78AECT.cpu0	ARM_Cortex-A78AE
ARMCortexA78AECT.cpu0.UTLB	TLB
ARMCortexA78AECT.cpu0.dtlb	TlbCadi
ARMCortexA78AECT.cpu0.l1dcache	PVCache
ARMCortexA78AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu0.l1icache	PVCache
ARMCortexA78AECT.cpu0.l1icache.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.dtlb	TlbCadi
ARMCortexA78AECT.cpu1.l1dcache	PVCache
ARMCortexA78AECT.cpu1.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu1.l1icache	PVCache
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	GICv3CPUInterfaceDecoder

This model has the following MTI trace components:

Table 3-306: ARMCortexA78AECT MTI instances

InstanceName	ComponentName
ARMCortexA78AECT	ARMv8Cluster
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

InstanceName	ComponentName
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
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
ARMCortexA78AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

ARMCortexA78AECT contains the following CADI targets:

- ARM_Cortex-A78AE
- Cluster_ARM_Cortex-A78AE
- PVCache
- TlbCadi

About ARMCortexA78AECT

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 DBGSRREQ are not implemented, but DBGWRUPREQ 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 `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](#).

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.

Ports for ARMCortexA78AECT

Table 3-307: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>AENDMP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>ASTARTMP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>broadcastatomic</code>	Signal	Slave	CHI defined pins.
<code>broadcastcachemaint</code>	Signal	Slave	ACE defined pins.
<code>broadcastouter</code>	Signal	Slave	ACE defined pins.
<code>broadcastpersist</code>	Signal	Slave	CHI defined pins.
<code>cfgend[4]</code>	Signal	Slave	This signal if for EE bit initialisation.
<code>cfgsdisable</code>	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
<code>cfgte[4]</code>	Signal	Slave	This signal provides default exception handling state.
<code>clk_in</code>	ClockSignal	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
<code>clusterid</code>	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
<code>clusterpmuirq</code>	Signal	Master	DynamlQ pmu irq
<code>CNTHPIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>CNTHVIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPNSIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPSIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>cntvalueb</code>	CounterInterface	Slave	Interface to SoC level counter module.
<code>CNTVIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>commirq[4]</code>	Signal	Master	Interrupt signal from debug communications channel.
<code>core_clk_in[4]</code>	ClockSignal	Slave	The clock signal connected to the <code>core_clk_in</code> port is used to determine the rate at which each core executes instructions.
<code>cpuporeset[4]</code>	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
<code>cryptodisable[4]</code>	Signal	Slave	Disable cryptography extensions after reset.
<code>cti[4]</code>	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.

Name	Protocol	Type	Description
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.
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.

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 A78AECT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTL.R.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`

Type

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_lock_step**Type**

bool

Default value

0x0

Whether the core is configured in Dual Core Lock Step mode (FEAT_DCLS).

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

If true, Accelerator Coherency Port is configured.

has_coherent_icache**Type**

bool

Default value

0x0

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

If true, additional AXI peripheral port is configured.

has_statistical_profiling**Type**

bool

Default value

0x1

Whether Statistical Based Profiling is implemented (FEAT_SPE).

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

Set whether I-cache has stateful implementation.

`l3cache-hit_latency`**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

`periph_address_end`**Type**

int

Default value

0x0

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

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

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 3-308: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA78CCT

This model has the following Iris instances:

Table 3-309: 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.dtlb	TlbCadi
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

This model has the following MTI trace components:

Table 3-310: ARMCortexA78CCT MTI instances

InstanceName	ComponentName
ARMCortexA78CCT	ARMv8Cluster
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

ARMCortexA78CCT contains the following CADI targets:

- ARM_Cortex-A78C
- Cluster_ARM_Cortex-A78C
- PVCache
- TlbCadi

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 DBGPWRDUP and DBGRSTREQ are not implemented, but DBGPWRUPREQ 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 (0-7).

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](#).

Ports for ARMCortexA78CCT

Table 3-311: 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).

Name	Protocol	Type	Description
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.
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.
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.

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.
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.
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 ARMCortexA78CCT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

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

Number of cores per cluster.

cpi_div

Type

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul

Type

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`cpuX.l2cache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

If true, Accelerator Coherency Port is configured.

has_coherent_icache**Type**

bool

Default value

0x0

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether l-cache has stateful implementation.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end

Type

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start

Type

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency

Type

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.32 ARM CortexA78CT

ARM CortexA78CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-312: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM CortexA78CT

This model has the following Iris instances:

Table 3-313: ARM CortexA78CT Iris instances

InstanceName	ComponentName
ARM CortexA78CT	Cluster_ARM_Cortex-A78
ARM CortexA78CT.AMU	PVBusLogger
ARM CortexA78CT.AMU.mapper	PVBusMapper
ARM CortexA78CT.DAP	PVBusLogger
ARM CortexA78CT.DAP.mapper	PVBusMapper
ARM CortexA78CT.DSU	DSU
ARM CortexA78CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM CortexA78CT.DSU.mpam_busslave	PVBusSlave
ARM CortexA78CT.DSU.shared_cache	PVCache
ARM CortexA78CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARM CortexA78CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARM CortexA78CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARM CortexA78CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARM CortexA78CT.MMAP	PVBusLogger
ARM CortexA78CT.MMAP.mapper	PVBusMapper
ARM CortexA78CT.RAS	PVBusLogger
ARM CortexA78CT.RAS.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA78CT.cpu0	ARM_Cortex-A78
ARMCortexA78CT.cpu0.UTLB	TLB
ARMCortexA78CT.cpu0.dtlb	TlbCadi
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

This model has the following MTI trace components:

Table 3-314: ARMCortexA78CT MTI instances

InstanceName	ComponentName
ARMCortexA78CT	ARMv8Cluster
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
ARMCortexA78CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78CT.cpu0.l1icache	PVCache

InstanceName	ComponentName
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	GCv3CPUInterfaceDecoder

ARMCortexA78CT contains the following CADI targets:

- ARM_Cortex-A78
- Cluster_ARM_Cortex-A78
- PVCache
- TlbCadi

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](#).

Ports for ARMCortexA78CT

Table 3-315: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>AENDMP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>ASTARTMP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>broadcastatomic</code>	Signal	Slave	CHI defined pins.
<code>broadcastcachemaint</code>	Signal	Slave	ACE defined pins.
<code>broadcastouter</code>	Signal	Slave	ACE defined pins.
<code>broadcastpersist</code>	Signal	Slave	CHI defined pins.
<code>cfgend[4]</code>	Signal	Slave	This signal is for EE bit initialisation.
<code>cfgsdisable</code>	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
<code>cfgte[4]</code>	Signal	Slave	This signal provides default exception handling state.
<code>clk_in</code>	ClockSignal	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
<code>clusterid</code>	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
<code>clusterpmuirq</code>	Signal	Master	DynamlQ pmu irq
<code>CNTHPIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>CNTHVIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPNSIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPSIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>cntvalueb</code>	CounterInterface	Slave	Interface to SoC level counter module.
<code>CNTVIRQ[4]</code>	Signal	Master	Timer signals to SOC.
<code>commirq[4]</code>	Signal	Master	Interrupt signal from debug communications channel.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

If true, Accelerator Coherency Port is configured.

has_coherent_icache**Type**

bool

Default value

0x0

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

`icache-state_modelled`

Type

bool

Default value

0x0

Set whether I-cache has stateful implementation.

`l3cache-hit_latency`

Type

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end

Type

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start

Type

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency

Type

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency

Type

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.33 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 3-316: 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 23.

Changes in 11.27.17

Table 3-317: Model quality changes

From	To
rOp0=rel	rel
r1p3=rel	

Iris and MTI instances for ARMCortexA510CT

This model has the following Iris instances:

Table 3-318: 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
ARMCortexA510CT.RAS.mapper	PVBusMapper
ARMCortexA510CT.cpu0	ARM_Cortex-A510
ARMCortexA510CT.cpu0.UTLB	TLB
ARMCortexA510CT.cpu0.dtlb	TlbCadi
ARMCortexA510CT.cpu0.l1dcache	PVCache
ARMCortexA510CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT.cpu0.l1icache	PVCache
ARMCortexA510CT.cpu0.l1icache.upstream[0]	PVBusSlave

InstanceName	ComponentName
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.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-319: ARMCortexA510CT MTI instances

InstanceName	ComponentName
ARMCortexA510CT	ARMv8Cluster
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
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.l1icache	PVCache
ARMCortexA510CT.cpu0.l1icache.upstream[0]	PVBusSlave

InstanceName	ComponentName
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 contains the following CADI targets:

- ARM_Cortex-A510
- Cluster_ARM_Cortex-A510
- PVCache
- TlbCadi

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](#).

Ports for ARMCortexA510CT

Table 3-320: 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.

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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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.
CNTHVSIHQ[12]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[12]	Signal	Master	Timer signals to SOC.
CNTPSIHQ[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 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.
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[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.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).

Name	Protocol	Type	Description
vsei[12]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM Cortex A510CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

Value of CPU Configuration Register.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

bus_type**Type**

int

Default value

0x0

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

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's 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

Divider for calculating CPI (Cycles Per Instruction).

`cpi_mul`**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

`cpuX.CFGEND`**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

`cpuX.CRYPTODISABLE`**Type**

bool

Default value

0x0

Disable cryptographic features.

`cpuX.RVBARADDR`**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

cpuX.l2cache-read_latency**Type**

int

Default value

0x0

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

cpuX.l2cache-write_latency**Type**

int

Default value

0x0

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`**Type**

int

Default value

0x0

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

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

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

Number of claim tags.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

has_ete**Type**

bool

Default value

0x0

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

If true, an additional AXI peripheral port is configured.

icache-hit_latency

Type

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`

Type

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

`icache-state_modelled`

Type

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-has_mpam**Type**

bool

Default value

0x1

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency

Type

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

l3cache-write_access_latency**Type**

int

Default value

0x0

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

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

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: EL0 supports A32. This parameter is ignored in Rev0 i.e. when revision_number = 0.

memory_tagging_support_level**Type**

int

Default value

0x3

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

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

Number of transport nodes. Zero implies direct-connect configuration.

patch_level

Type

int

Default value

0x3

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

ras_pfg_clock_mhz**Type**

int

Default value

0xc

RAS Pseudo-Fault generation clock rate in MHz.

revision_number**Type**

int

Default value

0x1

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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_PAAuth traps.

walk_cache_latency

Type

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.34 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 3-321: 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 23.

Iris and MTI instances for ARMCortexA510CT_CortexA710CT

This model has the following Iris instances:

Table 3-322: ARMCortexA510CT_CortexA710CT Iris instances

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT	Cluster_ARM_Cortex510_CortexA710_Heterogeneous
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

InstanceName	ComponentName
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.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
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	TlbCadi
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.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

InstanceName	ComponentName
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	TlbCadi
ARMCortexA510CT_CortexA710CT.subcluster1.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-323: ARMCortexA510CT_CortexA710CT MTI instances

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT	ARMv8Cluster
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.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

InstanceName	ComponentName
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.l1icache	PVCache
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l1icache.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.l1icache	PVCache
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l1icache.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 contains the following CADI targets:

- ARM_Cortex-A510
- ARM_Cortex-A710
- Cluster_ARM_Cortex510_CortexA710_Heterogeneous
- PVCache
- Subcluster_ARM_Cortex-A510
- Subcluster_ARM_Cortex-A710
- TlbCadi

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`.



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](#).

Ports for ARMCortexA510CT_CortexA710CT

Table 3-324: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>AEND0MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND1MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND2MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND3MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>ASTART0MP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>ASTART1MP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>ASTART2MP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>ASTART3MP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>broadcastatomic</code>	Signal	Slave	CHI defined pins.
<code>broadcastcachemaint</code>	Signal	Slave	ACE defined pins.
<code>broadcastouter</code>	Signal	Slave	ACE defined pins.
<code>broadcastpersist</code>	Signal	Slave	CHI defined pins.
<code>cfgend[22]</code>	Signal	Slave	This signal is for EE bit initialisation.
<code>cfgsdisable</code>	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.

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.
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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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.
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_pcs_m_pchannel[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.
dbgprupreq[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.

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[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.
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.

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[22]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM Cortex A510CT_Cortex A710CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

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

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 PPUs 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

Set whether D-cache has stateful implementation.

default_opmode**Type**

int

Default value

0x4

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

Enable DynamlQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

If true, Accelerator Coherency Port is configured.

has_peripheral_port**Type**

bool

Default value

0x0

If true, an additional AXI peripheral port is configured.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-has_mpam**Type**

bool

Default value

0x0

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency

Type

int

Default value

0x0

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

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

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

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

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

Number of transport nodes. Zero implies direct-connect configuration.

pchannel_treat_simreset_as_poreset**Type**

bool

Default value

0x0

Register core as ON state to cluster with simulation reset.

periph_address_end**Type**

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

subcluster0.CPUCFR**Type**

int

Default value

0x0

Value of CPU Configuration Register.

subcluster0.NUM_CORES**Type**

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

subcluster0.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster0.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster0.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster0.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster0.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster0.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster0.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-read_latency**Type**

int

Default value

0x0

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

L2 Cache size in bytes.

subcluster0.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

subcluster0.cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-write_latency**Type**

int

Default value

0x0

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster0.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster0.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster0.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster0.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster0.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster0.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster0.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster0.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster0.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster0.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

Number of claim tags.

subcluster0.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster0.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster0.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster0.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster0.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster0.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster0.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster0.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster0.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster0.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster0.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster0.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster0.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster0.ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

subcluster0.has_ete**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster0.max_32bit_el**Type**

int

Default value

0x0

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: EL0 supports A32. This parameter is ignored in Rev0 i.e. when `revision_number = 0`.

subcluster0.memory_tagging_support_level**Type**

int

Default value

0x3

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.ras_pfg_clock_mhz**Type**

int

Default value

0xc

RAS Pseudo-Fault generation clock rate in MHz.

subcluster0.revision_number**Type**

int

Default value

0x1

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.NUM_CORES**Type**

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

subcluster1.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster1.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster1.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster1.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster1.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster1.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster1.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster1.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster1.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster1.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster1.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster1.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster1.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster1.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster1.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster1.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

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

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

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

L1 D-Cache size in bytes.

`subcluster1.dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

Number of claim tags.

subcluster1.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster1.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster1.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster1.ete.PIDR_REVAND**Type**

int

Default value

0x1

TRCPIDR REVAND value.

subcluster1.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster1.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster1.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster1.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster1.ete.REVISION**Type**

int

Default value

0x2

TRCIDR1 revision value.

subcluster1.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster1.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster1.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster1.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster1.etc.TRCRSRTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCRSR.TA value for a forcibly traced exception.

subcluster1.ext_abort_device_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE read external aborts.

subcluster1.ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

subcluster1.ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

subcluster1.ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster1.memory_tagging_support_level**Type**

int

Default value

0x2

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.35 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 3-325: IP revisions support

Revision	Quality level
CortexA510 r0p0	Full support
CortexA510 r1p3	Full support
CortexA710 r2p0	Full support
CortexX2 r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA510CT_CortexA710CT_CortexX2CT

This model has the following Iris instances:

Table 3-326: 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

InstanceName	ComponentName
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	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	TlbCadi
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l1icache	PVCache

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l1icache.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.l1icache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l1icache.upstream[0]	PVBusSlave
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	TlbCadi
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.l1icache	PVCache

InstanceName	ComponentName
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	TlbCadi
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-327: ARMCortexA510CT_CortexA710CT_CortexX2CT MTI instances

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX2CT	ARMv8Cluster
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

InstanceName	ComponentName
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	GCv3CPUInterfaceDecoder
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
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.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 contains the following CADI targets:

- ARM_Cortex-A510
- ARM_Cortex-A710
- ARM_Cortex-X2

- Cluster_ARM_CortexA510_CortexA710_CortexX2_Heterogeneous
- PVCache
- Subcluster_ARM_Cortex-A510
- Subcluster_ARM_Cortex-A710
- Subcluster_ARM_Cortex-X2
- TlbCadi

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.



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](#).

Ports for ARMCortexA510CT_CortexA710CT_CortexX2CT

Table 3-328: 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.
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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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.

Name	Protocol	Type	Description
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_pcsmpchannel[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.
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 PPUs 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.

Name	Protocol	Type	Description
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.
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_CortexX2CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMPO input signal).

AEND1_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

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 PPUs 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

Set whether D-cache has stateful implementation.

default_opmode**Type**

int

Default value

0x4

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

If true, Accelerator Coherency Port is configured.

has_peripheral_port**Type**

bool

Default value

0x0

If true, an additional AXI peripheral port is configured.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-has_mpam**Type**

bool

Default value

0x0

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Number of transport nodes. Zero implies direct-connect configuration.

pchannel_treat_simreset_as_poreset

Type

bool

Default value

0x0

Register core as ON state to cluster with simulation reset.

periph_address_end

Type

int

Default value

0x0

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

periph_address_start

Type

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

subcluster0.CPUCFR**Type**

int

Default value

0x0

Value of CPU Configuration Register.

subcluster0.NUM_CORES**Type**

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

subcluster0.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster0.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster0.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster0.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster0.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster0.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster0.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-read_latency**Type**

int

Default value

0x0

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

L2 Cache size in bytes.

subcluster0.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

subcluster0.cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-write_latency**Type**

int

Default value

0x0

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster0.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster0.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster0.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster0.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster0.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster0.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster0.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster0.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster0.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

`subcluster0.dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

Number of claim tags.

subcluster0.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster0.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster0.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster0.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster0.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster0.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster0.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster0.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster0.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster0.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster0.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster0.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster0.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster0.ete.TRCSRTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRTA value for a forcibly traced exception.

subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

subcluster0.has_ete**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster0.max_32bit_el**Type**

int

Default value

0x0

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: EL0 supports A32. This parameter is ignored in Rev0 i.e. when `revision_number` = 0.

`subcluster0.memory_tagging_support_level`**Type**

int

Default value

0x3

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

`subcluster0.ras_pfg_clock_mhz`**Type**

int

Default value

0xc

RAS Pseudo-Fault generation clock rate in MHz.

`subcluster0.revision_number`**Type**

int

Default value

0x1

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT_PAAuth 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.

subcluster0.walk_cache_latency**Type**

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.NUM_CORES**Type**

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

subcluster1.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster1.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster1.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster1.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster1.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster1.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster1.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`subcluster1.cpuX.l2cache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster1.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster1.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster1.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster1.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster1.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster1.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster1.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster1.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster1.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

Number of claim tags.

subcluster1.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster1.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster1.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster1.ete.PIDR_REVAND**Type**

int

Default value

0x1

TRCPIDR REVAND value.

subcluster1.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster1.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster1.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster1.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster1.ete.REVISION**Type**

int

Default value

0x2

TRCIDR1 revision value.

subcluster1.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster1.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster1.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster1.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster1.ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

subcluster1.ext_abort_device_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE read external aborts.

subcluster1.ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

subcluster1.ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

subcluster1.ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster1.memory_tagging_support_level**Type**

int

Default value

0x2

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.NUM_CORES**Type**

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

subcluster2.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster2.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster2.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster2.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster2.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster2.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster2.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster2.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster2.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster2.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster2.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster2.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster2.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster2.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster2.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster2.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster2.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster2.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

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

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

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

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

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

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

Number of claim tags.

subcluster2.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster2.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster2.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster2.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster2.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster2.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster2.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster2.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster2.ete.REVISION**Type**

int

Default value

0x2

TRCIDR1 revision value.

subcluster2.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster2.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster2.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster2.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster2.ete.TRCRSRTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCRSR.TA value for a forcibly traced exception.

subcluster2.ext_abort_device_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE read external aborts.

subcluster2.ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

subcluster2.ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

subcluster2.ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

subcluster2.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

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

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

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

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.memory_tagging_support_level**Type**

int

Default value

0x2

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.tlbi_stall_enabled

Type

bool

Default value

0x0

If true, tlbi 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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.36 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 3-329: IP revisions support

Revision	Quality level
CortexA510 r0p0	Full support
CortexA510 r1p3	Full support
CortexA710 r2p0	Full support

Revision	Quality level
CortexX3 r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA510CT_CortexA710CT_CortexX3CT

This model has the following Iris instances:

Table 3-330: 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.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
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	TlbCadi
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
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.sve	ScalableVectorExtension

InstanceName	ComponentName
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	TlbCadi
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	TlbCadi

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-331: ARMCortexA510CT_CortexA710CT_CortexX3CT MTI instances

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX3CT	ARMv8Cluster
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

InstanceName	ComponentName
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.l1icache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l1icache.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
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

ARMCortexA510CT_CortexA710CT_CortexX3CT contains the following CADI targets:

- ARM_Cortex-A510
- ARM_Cortex-A710
- ARM_Cortex-X3
- Cluster_ARM_CortexA510_CortexA710_CortexX3_Heterogeneous
- PVCache
- Subcluster_ARM_Cortex-A510
- Subcluster_ARM_Cortex-A710
- Subcluster_ARM_Cortex-X3
- TlbCadi

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](#).

Ports for ARMCortexA510CT_CortexA710CT_CortexX3CT

Table 3-332: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>AEND0MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND1MP</code>	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[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_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
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[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.

Name	Protocol	Type	Description
core_pcsmpchannel[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
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.

Name	Protocol	Type	Description
pvbush_m0	PVBus	Master	The core will generate bus requests on this port.
pvbush_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 ARM CortexA510CT_CortexA710CT_CortexX3CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

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

Set whether D-cache has stateful implementation.

default_opmode**Type**

int

Default value

0x4

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

If true, Accelerator Coherency Port is configured.

has_peripheral_port**Type**

bool

Default value

0x0

If true, an additional AXI peripheral port is configured.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-has_mpam**Type**

bool

Default value

0x0

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Number of transport nodes. Zero implies direct-connect configuration.

pchannel_treat_simreset_as_poreset

Type

bool

Default value

0x0

Register core as ON state to cluster with simulation reset.

periph_address_end

Type

int

Default value

0x0

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

periph_address_start

Type

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

subcluster0.CPUCFR**Type**

int

Default value

0x0

Value of CPU Configuration Register.

subcluster0.NUM_CORES**Type**

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

subcluster0.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster0.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster0.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster0.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster0.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster0.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster0.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-read_latency**Type**

int

Default value

0x0

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

L2 Cache size in bytes.

subcluster0.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

subcluster0.cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-write_latency**Type**

int

Default value

0x0

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster0.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster0.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster0.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster0.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster0.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster0.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster0.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster0.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster0.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

`subcluster0.dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

Number of claim tags.

subcluster0.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster0.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster0.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster0.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster0.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster0.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster0.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster0.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster0.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster0.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster0.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster0.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster0.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster0.ete.TRCSRTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRTA value for a forcibly traced exception.

subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

subcluster0.has_ete**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster0.max_32bit_el**Type**

int

Default value

0x0

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: EL0 supports A32. This parameter is ignored in Rev0 i.e. when `revision_number = 0`.

`subcluster0.memory_tagging_support_level`**Type**

int

Default value

0x3

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

`subcluster0.ras_pfg_clock_mhz`**Type**

int

Default value

0xc

RAS Pseudo-Fault generation clock rate in MHz.

`subcluster0.revision_number`**Type**

int

Default value

0x1

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT_PAAuth 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.

subcluster0.walk_cache_latency**Type**

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.NUM_CORES**Type**

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

subcluster1.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster1.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster1.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster1.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster1.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster1.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster1.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`subcluster1.cpuX.l2cache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster1.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster1.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster1.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster1.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster1.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster1.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster1.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster1.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster1.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

Number of claim tags.

subcluster1.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster1.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster1.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster1.ete.PIDR_REVAND**Type**

int

Default value

0x1

TRCPIDR REVAND value.

subcluster1.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster1.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster1.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster1.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster1.ete.REVISION**Type**

int

Default value

0x2

TRCIDR1 revision value.

subcluster1.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster1.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster1.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster1.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster1.ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

subcluster1.ext_abort_device_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE read external aborts.

subcluster1.ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

subcluster1.ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

subcluster1.ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster1.memory_tagging_support_level**Type**

int

Default value

0x2

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.CPUCFR**Type**

int

Default value

0x0

Value of CPU Configuration Register.

subcluster2.NUM_CORES**Type**

int

Default value

0x1

Number of cores per cluster.

subcluster2.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster2.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster2.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster2.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster2.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster2.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster2.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster2.cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

subcluster2.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster2.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster2.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster2.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster2.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster2.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster2.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster2.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster2.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster2.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster2.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster2.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster2.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

`subcluster2.dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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}],"
```

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

Number of claim tags.

subcluster2.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster2.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster2.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster2.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster2.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster2.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster2.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster2.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster2.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster2.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster2.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster2.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster2.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster2.ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

subcluster2.ext_abort_normal_noncacheable_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of normal noncacheable-read external aborts.

subcluster2.ext_abort_so_write_ras_type**Type**

int

Default value

0x2

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

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

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

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

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

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

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

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

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster2.instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

subcluster2.memory_tagging_support_level**Type**

int

Default value

0x3

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

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}]]"
```

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.stage12_tlb_size

Type

int

Default value

0x80

Number of stage1+2 tlb entries.

subcluster2.tlb_latency

Type

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.tlbi_stall_enabled

Type

bool

Default value

0x0

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

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.37 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 3-333: 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 23.

Iris and MTI instances for ARM CortexA510CT_CortexA715CT_CortexX3CT

This model has the following Iris instances:

Table 3-334: ARM CortexA510CT_CortexA715CT_CortexX3CT Iris instances

InstanceName	ComponentName
ARM CortexA510CT_CortexA715CT_CortexX3CT	Cluster_ARM_CortexA510_CortexA715_CortexX3_Heterogeneous
ARM CortexA510CT_CortexA715CT_CortexX3CT.AMU	PVBusLogger
ARM CortexA510CT_CortexA715CT_CortexX3CT.AMU.mapper	PVBusMapper
ARM CortexA510CT_CortexA715CT_CortexX3CT.DAP	PVBusLogger
ARM CortexA510CT_CortexA715CT_CortexX3CT.DAP.mapper	PVBusMapper
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU	DSU
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_cluster	PPUv1
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core0	PPUv1
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core0.busslave	PVBusSlave
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core1	PPUv1
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core1.busslave	PVBusSlave
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core2	PPUv1
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core2.busslave	PVBusSlave
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.mpam_busslave	PVBusSlave
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache	PVCache
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARM CortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[4]	PVBusSlave

InstanceName	ComponentName
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	GlCv3CPUInterfaceDecoder
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	TlbCadi
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
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.subcluster0.sve	ScalableVectorExtension
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1	Subcluster_ARM_Cortex-A715

InstanceName	ComponentName
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	TlbCadi
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	TlbCadi
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-335: ARMCortexA510CT_CortexA715CT_CortexX3CT MTI instances

InstanceName	ComponentName
ARMCortexA510CT_CortexA715CT_CortexX3CT	ARMv8Cluster
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

InstanceName	ComponentName
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
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

ARMCortexA510CT_CortexA715CT_CortexX3CT contains the following CADI targets:

- ARM_Cortex-A510
- ARM_Cortex-A715
- ARM_Cortex-X3
- Cluster_ARM_CortexA510_CortexA715_CortexX3_Heterogeneous
- PVCache
- Subcluster_ARM_Cortex-A510
- Subcluster_ARM_Cortex-A715
- Subcluster_ARM_Cortex-X3
- TlbCadi

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](#).

Ports for ARMCortexA510CT_CortexA715CT_CortexX3CT

Table 3-336: 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[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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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_CortexA715CT_CortexX3CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

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

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 PPUs 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

Set whether D-cache has stateful implementation.

default_opmode**Type**

int

Default value

0x4

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

If true, an additional AXI peripheral port is configured.

icache-state_modelled

Type

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-has_mpam

Type

bool

Default value

0x0

L3 Cache has MPAM support.

l3cache-hit_latency

Type

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

l3cache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Number of transport nodes. Zero implies direct-connect configuration.

subcluster0.CPUCFR

Type

int

Default value

0x0

Value of CPU Configuration Register.

subcluster0.NUM_CORES

Type

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

subcluster0.cpi_div

Type

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster0.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster0.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster0.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster0.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster0.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter `trace_special_hlt_imm16`.

`subcluster0.cpuX.l2cache-hit_latency`

Type

int

Default value

0x0

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

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

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

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

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-read_latency**Type**

int

Default value

0x0

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

L2 Cache size in bytes.

subcluster0.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

subcluster0.cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-write_latency**Type**

int

Default value

0x0

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster0.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster0.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster0.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster0.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster0.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster0.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster0.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster0.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster0.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster0.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

Number of claim tags.

`subcluster0.etc.MAX_INST_PER_Q`**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

`subcluster0.etc.NumberOfRSPairs`**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster0.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster0.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster0.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster0.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster0.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster0.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster0.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster0.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster0.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster0.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster0.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster0.ete.TRCRSRTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCRSR.TA value for a forcibly traced exception.

subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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 MPAMO_EL1, MPAM1_EL1, MPAM2_EL2, MPAM3_EL3, MPAMHCR_EL2.

subcluster0.has_coherent_icache

Type

bool

Default value

0x1

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

subcluster0.has_ete

Type

bool

Default value

0x0

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

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster0.max_32bit_el**Type**

int

Default value

0x0

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: EL0 supports A32. This parameter is ignored in Rev0 i.e. when `revision_number = 0`.

subcluster0.memory_tagging_support_level**Type**

int

Default value

0x3

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.ras_pfg_clock_mhz**Type**

int

Default value

0xc

RAS Pseudo-Fault generation clock rate in MHz.

subcluster0.revision_number

Type

int

Default value

0x1

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlbi_stall_enabled

Type

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.CPUCFR**Type**

int

Default value

0x0

Value of CPU Configuration Register.

subcluster1.NUM_CORES**Type**

int

Default value

0x1

Number of cores per cluster.

subcluster1.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster1.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster1.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster1.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster1.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster1.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster1.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster1.cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

subcluster1.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`subcluster1.cpuX.l2cache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster1.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster1.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster1.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster1.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster1.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster1.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster1.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster1.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster1.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster1.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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}],"
```

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

Number of claim tags.

subcluster1.ete.MAX_INST_PER_Q

Type

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster1.ete.NumberOfRSPairs

Type

int

Default value

0x8

Number of resource selector pairs.

subcluster1.ete.PIDR_CMOD

Type

int

Default value

0x0

TRCPIDR CMOD value.

subcluster1.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster1.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster1.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster1.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster1.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster1.ete.REVISION**Type**

int

Default value

0x1

TRCIDR1 revision value.

subcluster1.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster1.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster1.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster1.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster1.ete.TRCRSRTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCRSR.TA value for a forcibly traced exception.

subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

Whether Statistical Based Profiling is implemented (FEAT_SPE).

subcluster1.icache-hit_latency

Type

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster1.instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

subcluster1.memory_tagging_support_level**Type**

int

Default value

0x3

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

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}]]"
```

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.

subcluster1.ptw_latency

Type

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.stage12_tlb_size

Type

int

Default value

0x80

Number of stage1+2 tlb entries.

subcluster1.tlb_latency

Type

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlbi_stall_enabled**Type**

bool

Default value

0x0

If true, tlbi 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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.CPUCFR**Type**

int

Default value

0x0

Value of CPU Configuration Register.

subcluster2.NUM_CORES**Type**

int

Default value

0x1

Number of cores per cluster.

subcluster2.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster2.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster2.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster2.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster2.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster2.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster2.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster2.cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

subcluster2.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster2.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster2.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster2.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster2.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster2.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster2.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster2.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster2.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster2.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster2.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster2.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster2.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster2.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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}],"
```

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.etc.CLAIMTAGS**Type**

int

Default value

0x20

Number of claim tags.

subcluster2.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster2.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster2.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster2.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster2.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster2.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster2.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster2.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster2.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster2.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster2.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster2.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster2.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster2.ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

subcluster2.ext_abort_normal_noncacheable_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of normal noncacheable-read external aborts.

subcluster2.ext_abort_so_write_ras_type**Type**

int

Default value

0x2

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

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

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

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

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

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

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

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

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster2.instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

subcluster2.memory_tagging_support_level**Type**

int

Default value

0x3

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

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}]"
```

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

subcluster2.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.38 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 3-337: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA520AECT

This model has the following Iris instances:

Table 3-338: ARMCortexA520AECT Iris instances

InstanceName	ComponentName
ARMCortexA520AECT	Cluster_ARM_Cortex-A520AE
ARMCortexA520AECT.AMU	PVBusLogger
ARMCortexA520AECT.AMU.mapper	PVBusMapper
ARMCortexA520AECT.DAP	PVBusLogger
ARMCortexA520AECT.DAP.mapper	PVBusMapper
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.dtlb	TlbCadi
ARMCortexA520AECT.cpu0.l1dcache	PVCache
ARMCortexA520AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520AECT.cpu0.l1icache	PVCache
ARMCortexA520AECT.cpu0.l1icache.upstream[0]	PVBusSlave

InstanceName	ComponentName
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.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-339: ARMCortexA520AECT MTI instances

InstanceName	ComponentName
ARMCortexA520AECT	ARMv8Cluster
ARMCortexA520AECT.AMU	PVBusLogger
ARMCortexA520AECT.AMU.mapper	PVBusMapper
ARMCortexA520AECT.DAP	PVBusLogger
ARMCortexA520AECT.DAP.mapper	PVBusMapper
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.l1licache	PVCache

InstanceName	ComponentName
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	GlCv3CPUInterfaceDecoder

ARMCortexA520AECT contains the following CADI targets:

- ARM_Cortex-A520AE
- Cluster_ARM_Cortex-A520AE
- PVCache
- TlbCadi

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.

Ports for ARMCortexA520AECT

Table 3-340: 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).

Name	Protocol	Type	Description
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_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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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
coreerrirq[14]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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 ARMCortexA520AECT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Value of CPU Configuration Register.

DBGROMADDR**Type**

int

Default value

0x0

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x0

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

bus_type**Type**

int

Default value

0x0

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

cluster_split_lock_config**Type**

int

Default value

0x1

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_cache_protection**Type**

int

Default value

0x1

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"} }"
```

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

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 PPUs 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

Divider for calculating CPI (Cycles Per Instruction).

`cpi_mul`**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

`cpuX.CFGEND`**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

`cpuX.CRYPTODISABLE`**Type**

bool

Default value

0x0

Disable cryptographic features.

`cpuX.RVBARADDR`**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

`cpuX.enable_trace_special_hlt_imm16`**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

cpuX.l2cache-read_latency

Type

int

Default value

0x0

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency

Type

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

cpuX.l2cache-write_latency**Type**

int

Default value

0x0

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

Whether the core is configured in Dual Core Lock Step mode (FEAT_DCLS).

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Number of claim tags.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR_EL1[63:32].

has_dot_product**Type**

int

Default value

0x2

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

has_ete**Type**

bool

Default value

0x0

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

Support the Transient Fault Protection (TFP) flop parity errors through RAS registers (FEAT_TFP).

has_peripheral_port**Type**

bool

Default value

0x0

If true, an additional AXI peripheral port is configured.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether l-cache has stateful implementation.

l3cache-has_mpam**Type**

bool

Default value

0x1

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

l3cache-read_latency**Type**

int

Default value

0x0

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

l3cache-write_access_latency**Type**

int

Default value

0x0

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

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

l3cache-write_latency**Type**

int

Default value

0x0

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

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

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

Number of ACP ports.

num_nodes

Type

int

Default value

0x1

Number of transport nodes. Zero implies direct-connect configuration.

pmu-num_counters**Type**

int

Default value

0x6

Number of PMU counters implemented.

ptw_latency**Type**

int

Default value

0x0

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 } ]"
```

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

RAS Pseudo-Fault generation clock rate in MHz.

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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_PAAuth traps.

walk_cache_latency

Type

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.39 ARM Cortex A520CT

ARM Cortex A520CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-341: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM Cortex A520CT

This model has the following Iris instances:

Table 3-342: 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

InstanceName	ComponentName
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
ARMCortexA520CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT.RAS	PVBusLogger
ARMCortexA520CT.RAS.mapper	PVBusMapper
ARMCortexA520CT.cpu0	ARM_Cortex-A520
ARMCortexA520CT.cpu0.UTLB	TLB
ARMCortexA520CT.cpu0.dtlb	TlbCadi
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.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-343: ARMCortexA520CT MTI instances

InstanceName	ComponentName
ARMCortexA520CT	ARMv8Cluster
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

InstanceName	ComponentName
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
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

ARMCortexA520CT contains the following CADI targets:

- ARM_Cortex-A520
- Cluster_ARM_Cortex-A520
- PVCache
- TlbCadi

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](#).

Ports for ARMCortexA520CT

Table 3-344: 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[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 PPUs that informs thermal controller of the cluster power info
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[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

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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 ARMCortexA520CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Value of CPU Configuration Register.

DBGROMADDR**Type**

int

Default value

0x0

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x0

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

bus_type**Type**

int

Default value

0x0

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

core_cache_protection**Type**

int

Default value

0x1

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"}}}"
```

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

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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

cpuX.l2cache-read_latency**Type**

int

Default value

0x0

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

L2 Cache size in bytes.

`cpuX.l2cache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

`cpuX.l2cache-write_access_latency`**Type**

int

Default value

0x0

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

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

cpuX.l2cache-write_latency**Type**

int

Default value

0x0

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

`dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`

Type

int

Default value

0x0

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

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

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

Number of claim tags.

`ete.MAX_INST_PER_Q`**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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 MPAMO_EL1, MPAM1_EL1, MPAM2_EL2, MPAM3_EL3, MPAMHCR_EL2.

has_actlr2**Type**

bool

Default value

0x1

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR_EL1[63:32].

has_dot_product**Type**

int

Default value

0x2

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

has_ete**Type**

bool

Default value

0x0

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

If true, an additional AXI peripheral port is configured.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`

Type

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-has_mpam**Type**

bool

Default value

0x1

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

l3cache-read_latency**Type**

int

Default value

0x0

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

L3 Cache size in bytes.

`l3cache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

`l3cache-write_access_latency`

Type

int

Default value

0x0

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

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

l3cache-write_latency**Type**

int

Default value

0x0

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

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

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

Number of ACP ports.

num_nodes**Type**

int

Default value

0x1

Number of transport nodes. Zero implies direct-connect configuration.

pmu-num_counters**Type**

int

Default value

0x6

Number of PMU counters implemented.

ptw_latency**Type**

int

Default value

0x0

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, \"ERXMISC0_reset\": 0x2, \"ERXSTATUS_IERR_mask\": 0x300 ,
\"ERXMISC1_mask\": 0x0FF8700FFFF31f0f, \"ERXPFGCTL_reset\": 0x1000 } ]"
```

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

RAS Pseudo-Fault generation clock rate in MHz.

revision_number

Type

int

Default value

0x0

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.40 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 3-345: 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 23.

Iris and MTI instances for ARMCortexA520CT_CortexA720CT

This model has the following Iris instances:

Table 3-346: ARMCortexA520CT_CortexA720CT Iris instances

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT	Cluster_ARM_CortexA520_CortexA720_Heterogeneous
ARMCortexA520CT_CortexA720CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA720CT.AMU.mapper	PVBusMapper

InstanceName	ComponentName
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
ARMCortexA520CT_CortexA720CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.gic_cpuif_decoder_cluster	GiCv3CPUInterfaceDecoder
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	TlbCadi
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l1icache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave

InstanceName	ComponentName
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.l1icache	PVCache
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	TlbCadi
ARMCortexA520CT_CortexA720CT.subcluster1.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-347: ARMCortexA520CT_CortexA720CT MTI instances

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT	ARMv8Cluster
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

InstanceName	ComponentName
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
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

ARMCortexA520CT_CortexA720CT contains the following CADI targets:

- ARM_Cortex-A520
- ARM_Cortex-A720
- Cluster_ARM_CortexA520_CortexA720_Heterogeneous
- PVCache
- Subcluster_ARM_Cortex-A520

- Subcluster_ARM_Cortex-A720
- TlbCadi

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](#).

Ports for ARMCortexA520CT_CortexA720CT

Table 3-348: Ports

Name	Protocol	Type	Description
<code>acp_s[2]</code>	PVBus	Slave	AXI ACP slave port.
<code>AEND0MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND1MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND2MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND3MP</code>	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[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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x0

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x1

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

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"} }"
```

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

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 PPUs 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

Set whether D-cache has stateful implementation.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

If true, an additional AXI peripheral port is configured.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-has_mpam**Type**

bool

Default value

0x0

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

l3cache-read_latency**Type**

int

Default value

0x0

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

l3cache-write_access_latency**Type**

int

Default value

0x0

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

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

l3cache-write_latency**Type**

int

Default value

0x0

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

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

Number of transport nodes. Zero implies direct-connect configuration.

revision_number**Type**

int

Default value

0x0

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

Value of CPU Configuration Register.

subcluster0.NUM_CORES**Type**

int

Default value

0x1

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

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

Divider for calculating CPI (Cycles Per Instruction).

subcluster0.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster0.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster0.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster0.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster0.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster0.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

`subcluster0.cpuX.l2cache-read_latency`

Type

int

Default value

0x0

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

L2 Cache size in bytes.

`subcluster0.cpuX.l2cache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

subcluster0.cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-write_latency**Type**

int

Default value

0x0

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster0.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-T32_HLT

Type

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-Thumb_SVC

Type

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-cmd_line

Type

string

Default value

""

Command line available to semihosting calls.

subcluster0.cpuX.semihosting-cwd

Type

string

Default value

""

Base directory for semihosting file access.

subcluster0.cpuX.semihosting-enable

Type

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster0.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster0.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster0.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster0.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster0.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster0.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster0.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

Number of claim tags.

subcluster0.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster0.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster0.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster0.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster0.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster0.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster0.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster0.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster0.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster0.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster0.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster0.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster0.ete.TRCRSRTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCRSR.TA value for a forcibly traced exception.

subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR_EL1[63:32].

subcluster0.has_dot_product

Type

int

Default value

0x2

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

subcluster0.has_ete

Type

bool

Default value

0x0

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

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster0.memory_tagging_support_level**Type**

int

Default value

0x3

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

Number of PMU counters implemented.

subcluster0.ptw_latency**Type**

int

Default value

0x0

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 } ]"
```


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

RAS Pseudo-Fault generation clock rate in MHz.

subcluster0.revision_number

Type

int

Default value

0x0

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlbi_stall_enabled

Type

bool

Default value

0x0

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

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT_PAAuth 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_PAAuth traps.

subcluster0.walk_cache_latency**Type**

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.CPUCFR**Type**

int

Default value

0x0

Value of CPU Configuration Register.

subcluster1.NUM_CORES**Type**

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

subcluster1.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster1.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster1.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster1.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster1.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster1.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster1.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster1.cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

subcluster1.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster1.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster1.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster1.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster1.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster1.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster1.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster1.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster1.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster1.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster1.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

`subcluster1.dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

Number of claim tags.

subcluster1.ete.ETE_REVISION**Type**

int

Default value

0x1

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

Maximum limit for the number of instructions implied by a Q element.

subcluster1.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster1.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster1.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster1.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster1.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster1.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster1.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster1.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster1.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster1.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster1.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster1.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster1.ete.TRCRSRTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCRSR.TA value for a forcibly traced exception.

subcluster1.ete.TSMARK**Type**

bool

Default value

0x1

Whether timestamp markers are supported.

subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

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

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

Whether Statistical Based Profiling is implemented (FEAT_SPE).

subcluster1.icache-hit_latency

Type

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

`subcluster1.instruction_tlb_size`

Type

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

`subcluster1.memory_tagging_support_level`

Type

int

Default value

0x3

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

Number of PMU counters implemented.

subcluster1.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

subcluster1.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT_PAAuth 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_PAAuth traps.

subcluster1.walk_cache_latency**Type**

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.41 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 3-349: 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 23.

Iris and MTI instances for ARMCortexA520CT_CortexA720CT_CortexX4CT

This model has the following Iris instances:

Table 3-350: 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

InstanceName	ComponentName
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
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[8]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.MMAP	PVBusLogger

InstanceName	ComponentName
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	GLCv3CPUInterfaceDecoder
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	TlbCadi
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

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l1icache.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	TlbCadi
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
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	TlbCadi
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-351: ARMCortexA520CT_CortexA720CT_CortexX4CT MTI instances

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT_CortexX4CT	ARMv8Cluster
ARMCortexA520CT_CortexA720CT_CortexX4CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.DAP	PVBusLogger

InstanceName	ComponentName
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
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.l1icache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l1icache.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

InstanceName	ComponentName
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.l1icache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l1icache.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.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 contains the following CADI targets:

- ARM_Cortex-A520
- ARM_Cortex-A720
- ARM_Cortex-X4
- Cluster_ARM_CortexA520_CortexA720_CortexX4_Heterogeneous
- PVCache
- Subcluster_ARM_Cortex-A520
- Subcluster_ARM_Cortex-A720
- Subcluster_ARM_Cortex-X4
- TlbCadi

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](#).

Ports for ARMCortexA520CT_CortexA720CT_CortexX4CT

Table 3-352: Ports

Name	Protocol	Type	Description
<code>acp_s[2]</code>	PVBus	Slave	AXI ACP slave port.
<code>AEND0MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND1MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND2MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND3MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>ASTART0MP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>ASTART1MP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>ASTART2MP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>ASTART3MP</code>	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[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_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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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.
CNTHVSIQ[36]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[36]	Signal	Master	Timer signals to SOC.
CNTPSIQ[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_pcs_m_pchannel[36]	PChannel	Master	Core PCSM signals
core_powerdown_out[36]	Signal	Master	DEPRECATED - An output from PPU 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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x0

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x1

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

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"} }"
```

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

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 PPUs 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

Set whether D-cache has stateful implementation.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

If true, an additional AXI peripheral port is configured.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-has_mpam**Type**

bool

Default value

0x1

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

l3cache-read_latency**Type**

int

Default value

0x0

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

L3 Cache size in bytes.

`l3cache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

`l3cache-write_access_latency`

Type

int

Default value

0x0

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

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

l3cache-write_latency**Type**

int

Default value

0x0

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

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_nodes**Type**

int

Default value

0x1

Number of transport nodes. Zero implies direct-connect configuration.

revision_number**Type**

int

Default value

0x0

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

Value of CPU Configuration Register.

subcluster0.NUM_CORES**Type**

int

Default value

0x1

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

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

Divider for calculating CPI (Cycles Per Instruction).

subcluster0.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster0.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster0.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster0.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster0.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster0.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-read_latency**Type**

int

Default value

0x0

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

L2 Cache size in bytes.

subcluster0.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

subcluster0.cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-write_latency**Type**

int

Default value

0x0

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster0.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster0.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster0.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster0.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster0.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster0.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster0.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster0.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster0.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster0.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

Number of claim tags.

`subcluster0.etc.MAX_INST_PER_Q`**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

`subcluster0.etc.PIDR_CM0D`**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster0.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster0.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster0.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster0.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster0.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster0.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster0.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster0.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster0.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster0.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster0.ete.TRCRSRTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCRSR.TA value for a forcibly traced exception.

subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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_act1r2**Type**

bool

Default value

0x1

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR_EL1[63:32].

subcluster0.has_dot_product**Type**

int

Default value

0x2

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

subcluster0.has_ete**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster0.memory_tagging_support_level**Type**

int

Default value

0x3

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

Number of PMU counters implemented.

subcluster0.ptw_latency**Type**

int

Default value

0x0

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 } ]"
```

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

RAS Pseudo-Fault generation clock rate in MHz.

subcluster0.revision_number**Type**

int

Default value

0x0

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.CPUCFR**Type**

int

Default value

0x0

Value of CPU Configuration Register.

subcluster1.NUM_CORES**Type**

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

subcluster1.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster1.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster1.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster1.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster1.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster1.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster1.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster1.cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

subcluster1.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`subcluster1.cpuX.l2cache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster1.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster1.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster1.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster1.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster1.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster1.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster1.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster1.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster1.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

`subcluster1.dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

Number of claim tags.

subcluster1.ete.ETE_REVISION**Type**

int

Default value

0x1

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

Maximum limit for the number of instructions implied by a Q element.

subcluster1.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster1.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster1.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster1.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster1.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster1.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster1.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster1.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster1.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster1.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster1.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster1.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster1.ete.TRCRSRTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCRSR.TA value for a forcibly traced exception.

subcluster1.ete.TSMARK**Type**

bool

Default value

0x1

Whether timestamp markers are supported.

subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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 MPAMO_EL1, MPAM1_EL1, MPAM2_EL2, MPAM3_EL3, MPAMHCR_EL2.

subcluster1.has_enhanced_pan

Type

int

Default value

0x2

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

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

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

Whether Statistical Based Profiling is implemented (FEAT_SPE).

subcluster1.icache-hit_latency

Type

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster1.instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

subcluster1.memory_tagging_support_level**Type**

int

Default value

0x3

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

Number of PMU counters implemented.

subcluster1.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

subcluster1.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT_PAAuth 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_PAAuth traps.

subcluster1.walk_cache_latency

Type

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.CPUCFR

Type

int

Default value

0x0

Value of CPU Configuration Register.

subcluster2.NUM_CORES

Type

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

subcluster2.cpi_div

Type

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster2.cpi_mul

Type

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster2.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster2.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster2.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster2.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster2.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster2.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`subcluster2.cpuX.l2cache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

`subcluster2.cpuX.l2cache-write_access_latency`

Type

int

Default value

0x0

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster2.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster2.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster2.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster2.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster2.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster2.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster2.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster2.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster2.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster2.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster2.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

`subcluster2.dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

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

Number of claim tags.

subcluster2.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster2.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster2.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster2.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster2.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster2.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster2.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster2.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster2.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster2.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster2.ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

subcluster2.ext_abort_so_write_ras_type**Type**

int

Default value

0x2

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

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

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

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

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

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

Whether Statistical Based Profiling is implemented (FEAT_SPE).

subcluster2.has_v8_7_spe_inverted_filtering**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the inverting filtering by events.

subcluster2.has_v8_7_spe_previous_branch_target**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the optional branch target feature is implemented.

subcluster2.icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster2.instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

subcluster2.memory_tagging_support_level**Type**

int

Default value

0x3

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

MPAM Whether MPAMIDR_EL1.HAS_ALTSP bit is set or clear.

subcluster2.mpamidr_has_force_ns**Type**

int

Default value

0x0

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

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

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

Number of PMU counters implemented.

subcluster2.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

subcluster2.tcr_txsz_undersize_should_fault**Type**

bool

Default value

0x0

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

subcluster2.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.42 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 3-353: 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 23.

Iris and MTI instances for ARMCortexA520CT_CortexA725CT

This model has the following Iris instances:

Table 3-354: 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.ext_bus	PVBusLogger

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.gic_cpuif_decoder_cluster	GiCv3CPUInterfaceDecoder
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	TlbCadi
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.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.l1icache	PVCache
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
ARMCortexA520CT_CortexA725CT.subcluster1.cpu1.dtlb	TlbCadi
ARMCortexA520CT_CortexA725CT.subcluster1.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-355: ARMCortexA520CT_CortexA725CT MTI instances

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT	ARMv8Cluster
ARMCortexA520CT_CortexA725CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA725CT.AMU.mapper	PVBusMapper

InstanceName	ComponentName
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.ext_bus	PVBusLogger
ARMCortexA520CT_CortexA725CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
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.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.subcluster1.cpu0	ARM_Cortex-A725
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l1dcache	PVCache

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l1icache	PVCache
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

ARMCortexA520CT_CortexA725CT contains the following CADI targets:

- ARM_Cortex-A520
- ARM_Cortex-A725
- Cluster_ARM_Cortex-A520_CortexA725_Heterogeneous
- PVCache
- Subcluster_ARM_Cortex-A520
- Subcluster_ARM_Cortex-A725
- TlbCadi

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:

- [3.5.39 ARMCortexA520CT](#) on page 1859.
- [3.5.48 ARMCortexA725CT](#) on page 2269.

Ports for ARMCortexA520CT_CortexA725CT

Table 3-356: 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_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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[26]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[26]	Signal	Master	Timer signals to SOC

Name	Protocol	Type	Description
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 PPU 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.
dbgprupreq[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.

Name	Protocol	Type	Description
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.
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

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x0

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x1

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

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\"}} }
```

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

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

Set whether D-cache has stateful implementation.

enable_simulation_performance_optimizations

Type

bool

Default value

0x1

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

If true, an additional AXI peripheral port is configured.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-has_mpam**Type**

bool

Default value

0x1

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

l3cache-read_latency**Type**

int

Default value

0x0

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

l3cache-write_access_latency**Type**

int

Default value

0x0

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

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

l3cache-write_latency**Type**

int

Default value

0x0

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

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

Number of transport nodes. Zero implies direct-connect configuration.

revision_number**Type**

int

Default value

0x0

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

Value of CPU Configuration Register.

subcluster0.NUM_CORES**Type**

int

Default value

0x1

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

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

Divider for calculating CPI (Cycles Per Instruction).

subcluster0.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster0.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster0.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster0.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster0.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster0.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-read_latency**Type**

int

Default value

0x0

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

L2 Cache size in bytes.

subcluster0.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

subcluster0.cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-write_latency**Type**

int

Default value

0x0

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster0.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster0.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster0.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster0.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster0.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster0.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster0.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster0.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster0.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

`subcluster0.dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

Number of claim tags.

subcluster0.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster0.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster0.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster0.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster0.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster0.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster0.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster0.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster0.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster0.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster0.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster0.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster0.ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR_EL1[63:32].

subcluster0.has_dot_product**Type**

int

Default value

0x2

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

subcluster0.has_ete**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

L1 I-Cache size in bytes.

`subcluster0.memory_tagging_support_level`

Type

int

Default value

0x3

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

Number of PMU counters implemented.

subcluster0.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.ras_extra_configurations**Type**

string

Default value

```
"[ { \"Index\": 1, \"EXMISCO_mask\": 0xFFFFc0003fc3, \"EXMISC1_mask\":  
0x03F870003FF30f07, \"ERXPFGCTL_reset\": 0x1000 }, { \"Index\": 2, \"EXMISCO_mask  
\": 0xFFFFe007ffc0, \"EXMISCO_reset\": 0x2, \"ERXSTATUS_IERR_mask\": 0x300 ,  
\"EXMISC1_mask\": 0x0FF8700ffff31f0f, \"ERXPFGCTL_reset\": 0x1000 } ]"
```

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 EXMISCn masks - these are 64 bit masks covering the 64 bit registers EXMISCn_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, "ERXCTLR_EL1_mask": 0x0, "ERXCTLR_EL1_reset": 0x0}, {"Index": 1, "EXMISCO_mask": 0x0, "EXMISCO_reset": 0x0, "ERXSTATUS_IERR_mask": 0x300}].

subcluster0.ras_pfg_clock_mhz**Type**

int

Default value

0xc

RAS Pseudo-Fault generation clock rate in MHz.

subcluster0.revision_number**Type**

int

Default value

0x0

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT_PAAuth 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.

subcluster0.walk_cache_latency**Type**

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.CPUCFR**Type**

int

Default value

0x0

Value of CPU Configuration Register.

subcluster1.NUM_CORES**Type**

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

subcluster1.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster1.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster1.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster1.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster1.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster1.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster1.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster1.cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

subcluster1.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster1.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster1.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster1.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster1.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster1.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster1.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster1.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster1.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster1.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster1.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster1.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

Number of claim tags.

subcluster1.ete.ETE_REVISION**Type**

int

Default value

0x1

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

Maximum limit for the number of instructions implied by a Q element.

subcluster1.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster1.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster1.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster1.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster1.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster1.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster1.ete.RETSTACK**Type**

int

Default value

0x1

Return stack depth.

subcluster1.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster1.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster1.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster1.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster1.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster1.ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

subcluster1.ete.TSMARK**Type**

bool

Default value

0x1

Whether timestamp markers are supported.

subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

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

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

Whether Statistical Based Profiling is implemented (FEAT_SPE).

subcluster1.icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster1.instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

subcluster1.memory_tagging_support_level**Type**

int

Default value

0x3

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

Number of PMU counters implemented.

subcluster1.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

subcluster1.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.43 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 3-357: 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 23.

Iris and MTI instances for ARMCortexA520CT_CortexA725CT_CortexX925CT

This model has the following Iris instances:

Table 3-358: 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

InstanceName	ComponentName
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
ARMCortexA520CT_CortexA725CT_CortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
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	TlbCadi
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l1icache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache	PVCache

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
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	TlbCadi
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	TlbCadi
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-359: ARMCortexA520CT_CortexA725CT_CortexX925CT MTI instances

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT_CortexX925CT	ARMv8Cluster
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

ARMCortexA520CT_CortexA725CT_CortexX925CT contains the following CADI targets:

- ARM_Cortex-A520
- ARM_Cortex-A725
- ARM_Cortex-X925
- Cluster_ARM_Cortex-A520_CortexA725_CortexX925_Heterogeneous
- PVCache
- Subcluster_ARM_Cortex-A520
- Subcluster_ARM_Cortex-A725
- Subcluster_ARM_Cortex-X925
- TlbCadi

About ARM_CortexA520CT_CortexA725CT_CortexX925CT

The number of cores in each subcluster is configurable using the following parameters:

subcluster0.NUM_CORES

Possible values are 1-12 (ARM_CortexA520CT).

subcluster1.NUM_CORES

Possible values are 1-12 (ARM_CortexA725CT).

subcluster2.NUM_CORES

Possible values are 1-12 (ARM_CortexX925CT).

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:

- [3.5.39 ARM_CortexA520CT](#) on page 1859.

- [3.5.48 ARMCortexA725CT](#) on page 2269.
- [3.5.74 ARMCortexX925CT](#) on page 2808.

Ports for ARMCortexA520CT_CortexA725CT_CortexX925CT

Table 3-360: Ports

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AENDOMP	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).
ASTARTOMP	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
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[36]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[36]	Signal	Master	Timer signals to SOC
CNTHVIRQ[36]	Signal	Master	Timer signals to SOC.

Name	Protocol	Type	Description
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.
dbgprupreq[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.
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 PPUs 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.

Name	Protocol	Type	Description
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.
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

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

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

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

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

Set whether D-cache has stateful implementation.

default_opmode**Type**

int

Default value

0x4

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

Enable DynamlQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

If true, an additional AXI peripheral port is configured.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-has_mpam**Type**

bool

Default value

0x1

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

l3cache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Value of CPU Configuration Register.

subcluster0.NUM_CORES**Type**

int

Default value

0x1

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

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

Divider for calculating CPI (Cycles Per Instruction).

subcluster0.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster0.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster0.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster0.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster0.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster0.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-read_latency**Type**

int

Default value

0x0

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

L2 Cache size in bytes.

subcluster0.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

subcluster0.cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

subcluster0.cpuX.l2cache-write_latency**Type**

int

Default value

0x0

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster0.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster0.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster0.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster0.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster0.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster0.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster0.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster0.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster0.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

`subcluster0.dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

Number of claim tags.

subcluster0.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster0.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster0.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster0.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster0.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster0.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster0.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster0.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster0.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster0.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster0.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster0.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster0.ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR_EL1[63:32].

subcluster0.has_dot_product**Type**

int

Default value

0x2

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

subcluster0.has_ete**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

L1 I-Cache size in bytes.

`subcluster0.memory_tagging_support_level`**Type**

int

Default value

0x3

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

Number of PMU counters implemented.

subcluster0.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.ras_extra_configurations**Type**

string

Default value

```
"[ { \"Index\": 1, \"EXMISCO_mask\": 0xFFFFc0003fc3, \"EXMISCO1_mask\":  
0x03F870003FF30f07, \"ERXPFGCTL_reset\": 0x1000 }, { \"Index\": 2, \"EXMISCO_mask  
\": 0xFFFFe007ffc0, \"EXMISCO_reset\": 0x2, \"ERXSTATUS_IERR_mask\": 0x300 ,  
\"EXMISCO1_mask\": 0x0FF8700ffff31f0f, \"ERXPFGCTL_reset\": 0x1000 } ]"
```

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 EXMISCn masks - these are 64 bit masks covering the 64 bit registers EXMISCn_EL1. E.g. [{"Index": 0, "EXMISCO_mask": 0x0, "EXMISCO_reset": 0x0, "EXMISCO1_mask": 0x0, "EXMISCO1_reset": 0x0, "EXMISCO2_mask": 0x0, "EXMISCO2_reset": 0x0, "EXMISCO3_mask": 0x0, "EXMISCO3_reset": 0x0, "ERXCTLR_EL1_mask": 0x0, "ERXCTLR_EL1_reset": 0x0}, {"Index": 1, "EXMISCO_mask": 0x0, "EXMISCO_reset": 0x0, "ERXSTATUS_IERR_mask": 0x300}].

subcluster0.ras_pfg_clock_mhz**Type**

int

Default value

0xc

RAS Pseudo-Fault generation clock rate in MHz.

subcluster0.revision_number**Type**

int

Default value

0x0

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT_PAAuth 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.

subcluster0.walk_cache_latency**Type**

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.CPUCFR**Type**

int

Default value

0x0

Value of CPU Configuration Register.

subcluster1.NUM_CORES**Type**

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

subcluster1.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster1.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster1.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster1.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster1.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster1.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster1.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster1.cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

subcluster1.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster1.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster1.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster1.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster1.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster1.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster1.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster1.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster1.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster1.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster1.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster1.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

Number of claim tags.

subcluster1.ete.ETE_REVISION**Type**

int

Default value

0x1

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

Maximum limit for the number of instructions implied by a Q element.

subcluster1.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster1.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster1.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster1.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster1.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster1.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster1.ete.RETSTACK**Type**

int

Default value

0x1

Return stack depth.

subcluster1.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster1.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster1.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster1.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster1.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster1.ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

subcluster1.ete.TSMARK**Type**

bool

Default value

0x1

Whether timestamp markers are supported.

subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

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

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

Whether Statistical Based Profiling is implemented (FEAT_SPE).

subcluster1.icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster1.instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

subcluster1.memory_tagging_support_level**Type**

int

Default value

0x3

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

Number of PMU counters implemented.

subcluster1.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

subcluster1.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.CPUCFR**Type**

int

Default value

0x0

Value of CPU Configuration Register.

subcluster2.NUM_CORES**Type**

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

subcluster2.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster2.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster2.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster2.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster2.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster2.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster2.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster2.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster2.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

subcluster2.cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster2.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster2.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster2.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster2.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster2.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster2.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster2.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster2.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster2.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster2.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster2.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster2.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster2.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Number of claim tags.

subcluster2.ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster2.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster2.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster2.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster2.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster2.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster2.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster2.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster2.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster2.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster2.ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

subcluster2.ext_abort_so_write_ras_type**Type**

int

Default value

0x2

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

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

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

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 MPAMO_EL1, MPAM1_EL1, MPAM2_EL2, MPAM3_EL3, MPAMHCR_EL2.

subcluster2.has_enhanced_pan

Type

int

Default value

0x2

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

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

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

Whether Statistical Based Profiling is implemented (FEAT_SPE).

subcluster2.has_v8_7_spe_inverted_filtering**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the inverting filtering by events.

subcluster2.has_v8_7_spe_previous_branch_target**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the optional branch target feature is implemented.

subcluster2.icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster2.instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

subcluster2.memory_tagging_support_level**Type**

int

Default value

0x3

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

MPAM Whether MPAMIDR_EL1.HAS_ALTSP bit is set or clear.

subcluster2.mpamidr_has_force_ns**Type**

int

Default value

0x0

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

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

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

Number of PMU counters implemented.

subcluster2.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

subcluster2.tcr_txsz_undersize_should_fault**Type**

bool

Default value

0x0

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

subcluster2.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster2.tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.44 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 3-361: IP revisions support

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA710CT

This model has the following Iris instances:

Table 3-362: ARMCortexA710CT Iris instances

InstanceName	ComponentName
ARMCortexA710CT	Cluster_ARM_Cortex-A710
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
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.dtlb	TlbCadi

InstanceName	ComponentName
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.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-363: ARMCortexA710CT MTI instances

InstanceName	ComponentName
ARMCortexA710CT	ARMv8Cluster
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
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

InstanceName	ComponentName
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 contains the following CADI targets:

- ARM_Cortex-A710
- Cluster_ARM_Cortex-A710
- PVCache
- TlbCadi

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](#).

Ports for ARMCortexA710CT

Table 3-364: 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_pcs_m_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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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.
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.

Name	Protocol	Type	Description
gicreset	Signal	Master	An output from PPU's 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.
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 ARMCortexA710CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT

Type

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

core_power_on_by_default**Type**

bool

Default value

0x0

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 PPUs 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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Number of claim tags.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.PIDR_REVAND**Type**

int

Default value

0x1

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.REVISION**Type**

int

Default value

0x2

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCRSRTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCRSR.TA value for a forcibly traced exception.

ext_abort_device_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

If true, Accelerator Coherency Port is configured.

has_ete**Type**

bool

Default value

0x0

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

If true, an additional AXI peripheral port is configured.

icache-hit_latency

Type

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`

Type

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

`icache-state_modelled`

Type

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

`l3cache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

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

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

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

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

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

Number of transport nodes. Zero implies direct-connect configuration.

pchannel_treat_simreset_as_poreset**Type**

bool

Default value

0x0

Register core as ON state to cluster with simulation reset.

periph_address_end**Type**

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.45 **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 3-365: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA715CT

This model has the following Iris instances:

Table 3-366: ARMCortexA715CT Iris instances

InstanceName	ComponentName
ARMCortexA715CT	Cluster_ARM_Cortex-A715
ARMCortexA715CT.AMU	PVBusLogger
ARMCortexA715CT.AMU.mapper	PVBusMapper
ARMCortexA715CT.DAP	PVBusLogger
ARMCortexA715CT.DAP.mapper	PVBusMapper

InstanceName	ComponentName
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
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.dtlb	TlbCadi
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.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-367: ARMCortexA715CT MTI instances

InstanceName	ComponentName
ARMCortexA715CT	ARMv8Cluster
ARMCortexA715CT.AMU	PVBusLogger
ARMCortexA715CT.AMU.mapper	PVBusMapper
ARMCortexA715CT.DAP	PVBusLogger

InstanceName	ComponentName
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
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

ARMCortexA715CT contains the following CADI targets:

- ARM_Cortex-A715
- Cluster_ARM_Cortex-A715
- PVCache
- TlbCadi

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.

Ports for ARMCortexA715CT

Table 3-368: 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_pcs_m_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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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.
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.

Name	Protocol	Type	Description
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.
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).

Name	Protocol	Type	Description
vsei[12]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM Cortex A715CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

Value of CPU Configuration Register.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

bus_type**Type**

int

Default value

0x0

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

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

Divider for calculating CPI (Cycles Per Instruction).

`cpi_mul`**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

`cpuX.CFGEND`**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

`cpuX.CFGTE`**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

`cpuX.CRYPTODISABLE`**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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}],"
```

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

Number of claim tags.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.REVISION**Type**

int

Default value

0x1

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

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

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

If true, an additional AXI peripheral port is configured.

has_statistical_profiling

Type

bool

Default value

0x0

Whether Statistical Based Profiling is implemented (FEAT_SPE).

icache-hit_latency

Type

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`

Type

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

`icache-state_modelled`

Type

bool

Default value

0x0

Set whether I-cache has stateful implementation.

instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

l3cache-has_mpam**Type**

bool

Default value

0x1

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

l3cache-write_access_latency**Type**

int

Default value

0x0

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

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

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

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

Number of transport nodes. Zero implies direct-connect configuration.

pmu-num_counters**Type**

int

Default value

0x6

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}]"
```

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.46 ARM Cortex A720AECT

ARM Cortex A720AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-369: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM Cortex A720AECT

This model has the following Iris instances:

Table 3-370: ARM Cortex A720AECT Iris instances

InstanceName	ComponentName
ARM Cortex A720AECT	Cluster_ARM_Cortex-A720AE
ARM Cortex A720AECT.AMU	PVBusLogger
ARM Cortex A720AECT.AMU.mapper	PVBusMapper
ARM Cortex A720AECT.DAP	PVBusLogger
ARM Cortex A720AECT.DAP.mapper	PVBusMapper
ARM Cortex A720AECT.DSU	DSU
ARM Cortex A720AECT.DSU.PPU_cluster	PPUv1
ARM Cortex A720AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARM Cortex A720AECT.DSU.PPU_core0	PPUv1
ARM Cortex A720AECT.DSU.PPU_core0.busslave	PVBusSlave
ARM Cortex A720AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM Cortex A720AECT.DSU.mpam_busslave	PVBusSlave
ARM Cortex A720AECT.DSU.shared_cache	PVCache

InstanceName	ComponentName
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.dtlb	TlbCadi
ARMCortexA720AECT.cpu0.l1dcache	PVCache
ARMCortexA720AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA720AECT.cpu0.l1icache	PVCache
ARMCortexA720AECT.cpu0.l1icache.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.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-371: ARMCortexA720AECT MTI instances

InstanceName	ComponentName
ARMCortexA720AECT	ARMv8Cluster
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

InstanceName	ComponentName
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
ARMCortexA720AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

ARMCortexA720AECT contains the following CADI targets:

- ARM_Cortex-A720AE
- Cluster_ARM_Cortex-A720AE
- PVCache
- TlbCadi

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.

Ports for ARMCortexA720AECT

Table 3-372: 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_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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[14]	Signal	Master	Timer signals to SOC.

Name	Protocol	Type	Description
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.
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 PPUs 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.

Name	Protocol	Type	Description
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.
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

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

Value of CPU Configuration Register.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

bus_type**Type**

int

Default value

0x0

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

cluster_split_lock_config**Type**

int

Default value

0x1

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

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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`**Type**

int

Default value

0x0

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

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

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

Enable DynamlQ diagnostic messages.

enable_lock_step**Type**

bool

Default value

0x0

Whether the core is configured in Dual Core Lock Step mode (FEAT_DCLS).

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Number of claim tags.

ete.ETE_REVISION**Type**

int

Default value

0x1

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

ete.TSMARK**Type**

bool

Default value

0x1

Whether timestamp markers are supported.

force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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 MPAMO_EL1, MPAM1_EL1, MPAM2_EL2, MPAM3_EL3, MPAMHCR_EL2.

has_enhanced_pan**Type**

int

Default value

0x2

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

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

Support the Transient Fault Protection (TFP) flop parity errors through RAS registers (FEAT_TFP).

has_large_va**Type**

int

Default value

0x0

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

If true, an additional AXI peripheral port is configured.

has_statistical_profiling**Type**

bool

Default value

0x0

Whether Statistical Based Profiling is implemented (FEAT_SPE).

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

l3cache-has_mpam**Type**

bool

Default value

0x1

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

l3cache-write_access_latency**Type**

int

Default value

0x0

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

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

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

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

Number of transport nodes. Zero implies direct-connect configuration.

pmu-num_counters**Type**

int

Default value

0x6

Number of PMU counters implemented.

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

Number of stage1+2 tlb entries.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.47 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 3-373: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA720CT

This model has the following Iris instances:

Table 3-374: ARMCortexA720CT Iris instances

InstanceName	ComponentName
ARMCortexA720CT	Cluster_ARM_Cortex-A720
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

InstanceName	ComponentName
ARMCortexA720CT.cpu0.UTLB	TLB
ARMCortexA720CT.cpu0.dtlb	TlbCadi
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.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-375: ARMCortexA720CT MTI instances

InstanceName	ComponentName
ARMCortexA720CT	ARMv8Cluster
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

InstanceName	ComponentName
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

ARMCortexA720CT contains the following CADI targets:

- ARM_Cortex-A720
- Cluster_ARM_Cortex-A720
- PVCache
- TlbCadi

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

- COREINSTRET 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](#).

Ports for ARMCortexA720CT

Table 3-376: 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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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
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.

Name	Protocol	Type	Description
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.
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 ARMCortexA720CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Value of CPU Configuration Register.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

bus_type**Type**

int

Default value

0x0

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

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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`cpuX.l2cache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Number of claim tags.

ete.ETE_REVISION**Type**

int

Default value

0x1

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

ete.TSMARK**Type**

bool

Default value

0x1

Whether timestamp markers are supported.

force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

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

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

If true, an additional AXI peripheral port is configured.

has_statistical_profiling**Type**

bool

Default value

0x0

Whether Statistical Based Profiling is implemented (FEAT_SPE).

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

`icache-state_modelled`**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

`instruction_tlb_size`**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

`l3cache-has_mpam`**Type**

bool

Default value

0x1

L3 Cache has MPAM support.

`l3cache-hit_latency`**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency

Type

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

l3cache-write_access_latency

Type

int

Default value

0x0

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

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

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

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

Number of ACP ports.

num_nodes**Type**

int

Default value

0x1

Number of transport nodes. Zero implies direct-connect configuration.

pmu-num_counters**Type**

int

Default value

0x6

Number of PMU counters implemented.

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.48 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 3-377: IP revisions support

Revision	Quality level
rOp0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexA725CT

This model has the following Iris instances:

Table 3-378: ARMCortexA725CT Iris instances

InstanceName	ComponentName
ARMCortexA725CT	Cluster_ARM_Cortex-A725
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
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.dtlb	TlbCadi
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.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-379: ARMCortexA725CT MTI instances

InstanceName	ComponentName
ARMCortexA725CT	ARMv8Cluster
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
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	GICv3CPUInterfaceDecoder

ARMCortexA725CT contains the following CADI targets:

- ARM_Cortex-A725
- Cluster_ARM_Cortex-A725
- PVCache
- TlbCadi

Ports for ARMCortexA725CT

Table 3-380: 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_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
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[14]	Signal	Master	Timer signals to SOC.

Name	Protocol	Type	Description
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.
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 PPUs 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.

Name	Protocol	Type	Description
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.
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 ARMCortexA725CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMPO input signal).

AEND1_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

Value of CPU Configuration Register.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

bus_type**Type**

int

Default value

0x0

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

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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Number of claim tags.

ete.ETE_REVISION**Type**

int

Default value

0x1

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x1

Return stack depth.

ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

ete.TSMARK**Type**

bool

Default value

0x1

Whether timestamp markers are supported.

force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

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

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

If true, an additional AXI peripheral port is configured.

has_statistical_profiling**Type**

bool

Default value

0x0

Whether Statistical Based Profiling is implemented (FEAT_SPE).

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`

Type

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

`icache-state_modelled`

Type

bool

Default value

0x0

Set whether I-cache has stateful implementation.

instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

l3cache-has_mpam**Type**

bool

Default value

0x1

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

l3cache-write_access_latency**Type**

int

Default value

0x0

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

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

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

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.

pmu-num_counters**Type**

int

Default value

0x6

Number of PMU counters implemented.

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.49 **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 3-381: 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 23.

Iris and MTI instances for ARMCortexA725CT_CortexX925CT

This model has the following Iris instances:

Table 3-382: 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
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
ARMCortexA725CT_CortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
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	TlbCadi
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l1icache	PVCache

InstanceName	ComponentName
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
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
ARMCortexA725CT_CortexX925CT.subcluster1.cpu1.dtlb	TlbCadi
ARMCortexA725CT_CortexX925CT.subcluster1.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-383: ARMCortexA725CT_CortexX925CT MTI instances

InstanceName	ComponentName
ARMCortexA725CT_CortexX925CT	ARMv8Cluster
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

InstanceName	ComponentName
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
ARMCortexA725CT_CortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder
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.l1icache	PVCache
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l1icache.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.l1icache	PVCache
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 contains the following CADI targets:

- ARM_Cortex-A725
- ARM_Cortex-X925

- Cluster_ARM_CortexA725_CortexX925_Heterogeneous
- PVCache
- Subcluster_ARM_Cortex-A725
- Subcluster_ARM_Cortex-X925
- TlbCadi

About ARMCortexA725CT_CortexX925CT

The number of cores in each subcluster is configurable using the following parameters:

subcluster0.NUM_CORES

Possible values are 1-13 (ARMCortexA725CT).

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:

- [3.5.48 ARMCortexA725CT](#) on page 2269.
- [3.5.74 ARMCortexX925CT](#) on page 2808.

Ports for ARMCortexA725CT_CortexX925CT

Table 3-384: Ports

Name	Protocol	Type	Description
<code>acp_s[2]</code>	PVBus	Slave	AXI ACP slave port.
<code>AENDOMP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND1MP</code>	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[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 PPUs that informs thermal controller of the cluster power info
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[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.

Name	Protocol	Type	Description
core_pcsmpchannel[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
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.

Name	Protocol	Type	Description
pvbush_m0	PVBus	Master	The core will generate bus requests on this port.
pvbush_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.
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 CortexA725CT_CortexX925CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

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

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 PPUs 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

Set whether D-cache has stateful implementation.

default_opmode**Type**

int

Default value

0x4

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

If true, an additional AXI peripheral port is configured.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

l3cache-has_mpam**Type**

bool

Default value

0x1

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

l3cache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Value of CPU Configuration Register.

subcluster0.NUM_CORES

Type

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

subcluster0.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster0.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster0.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster0.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster0.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster0.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster0.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster0.cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

subcluster0.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster0.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster0.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster0.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster0.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster0.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster0.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster0.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster0.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster0.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster0.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster0.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster0.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster0.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

Number of claim tags.

`subcluster0.ete.ETE_REVISION`**Type**

int

Default value

0x1

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

Maximum limit for the number of instructions implied by a Q element.

subcluster0.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster0.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster0.ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

subcluster0.ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

subcluster0.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster0.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster0.ete.RETSTACK**Type**

int

Default value

0x1

Return stack depth.

subcluster0.ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

subcluster0.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster0.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster0.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster0.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster0.ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

subcluster0.ete.TSMARK**Type**

bool

Default value

0x1

Whether timestamp markers are supported.

subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

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

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

Whether Statistical Based Profiling is implemented (FEAT_SPE).

subcluster0.icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster0.instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

subcluster0.memory_tagging_support_level**Type**

int

Default value

0x3

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

Number of PMU counters implemented.

subcluster0.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

subcluster0.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster0.tlbi_stall_enabled**Type**

bool

Default value

0x0

If true, tlbi 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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.CPUCFR**Type**

int

Default value

0x0

Value of CPU Configuration Register.

subcluster1.NUM_CORES**Type**

int

Default value

0x1

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

subcluster1.cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

subcluster1.cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

subcluster1.cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

subcluster1.cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

subcluster1.cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

subcluster1.cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

subcluster1.cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

subcluster1.cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

subcluster1.cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

`subcluster1.cpuX.l2cache-write_access_latency`**Type**

int

Default value

0x0

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

subcluster1.cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

subcluster1.cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

subcluster1.cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

subcluster1.cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

subcluster1.cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

subcluster1.cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

subcluster1.cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

subcluster1.cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

subcluster1.cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

subcluster1.cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

subcluster1.dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 D-Cache size in bytes.

subcluster1.dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Number of claim tags.

`subcluster1.ete.MAX_INST_PER_Q`**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

subcluster1.ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

subcluster1.ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

subcluster1.ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

subcluster1.ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

subcluster1.ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

subcluster1.ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

subcluster1.ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

subcluster1.ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

subcluster1.ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

subcluster1.ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

subcluster1.ext_abort_so_write_ras_type**Type**

int

Default value

0x2

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

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

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

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

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

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

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

Whether Statistical Based Profiling is implemented (FEAT_SPE).

subcluster1.has_v8_7_spe_inverted_filtering**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the inverting filtering by events.

subcluster1.has_v8_7_spe_previous_branch_target**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the optional branch target feature is implemented.

subcluster1.icache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

L1 I-Cache size in bytes.

subcluster1.instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

subcluster1.memory_tagging_support_level**Type**

int

Default value

0x3

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

MPAM Whether MPAMIDR_EL1.HAS_ALTSP bit is set or clear.

subcluster1.mpamidr_has_force_ns**Type**

int

Default value

0x0

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

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.

subcluster1.mpamidr_has_tidr**Type**

int

Default value

0x1

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

Number of PMU counters implemented.

subcluster1.ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

subcluster1.tcr_txsz_undersize_should_fault**Type**

bool

Default value

0x0

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

subcluster1.tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

subcluster1.tlbi_stall_enabled

Type
bool

Default value
0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.50 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 3-385: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexM0CT

This model has the following Iris instances:

Table 3-386: 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 3-387: 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

ARMCortexM0CT contains the following CADI targets:

- ARM_Cortex-M0

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.

Ports for ARMCortexM0CT

Table 3-388: 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.

Name	Protocol	Type	Description
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 ARM Cortex M0CT

BIGENDINIT

Type

bool

Default value

0x0

Initialize processor to big endian mode.

BKPT

Type

int

Default value

0x4

Number of breakpoint unit comparators implemented.

DBG

Type

bool

Default value

0x1

Set whether debug extensions are implemented.

NUM_IRQ**Type**

int

Default value

0x20

Number of user interrupts.

SYST**Type**

bool

Default value

0x1

Enable support for SysTick timer functionality.

WIC**Type**

bool

Default value

0x1

Include support for WIC-mode deep sleep.

WPT**Type**

int

Default value

0x2

Number of watchpoint unit comparators implemented.

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit

Type
int

Default value
0x20700000

Virtual address of top of heap.

semihosting-stack_base

Type
int

Default value
0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit

Type
int

Default value
0x20700000

Virtual address of stack limit.

3.5.51 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 3-389: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexM0PlusCT

This model has the following Iris instances:

Table 3-390: ARMCortexM0PlusCT Iris instances

InstanceName	ComponentName
ARMCortexM0PlusCT	ARM_Cortex-M0+
ARMCortexM0PlusCT.acp_mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexM0PlusCT.ext_bus	PVBusLogger
ARMCortexM0PlusCT.ext_bus.mapper	PVBusMapper
ARMCortexM0PlusCT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-391: ARMCortexM0PlusCT MTI instances

InstanceName	ComponentName
ARMCortexM0PlusCT	ARM_Cortex-M0plus
ARMCortexM0PlusCT.acp_mapper	PVBusMapper
ARMCortexM0PlusCT.ext_bus	PVBusLogger
ARMCortexM0PlusCT.ext_bus.mapper	PVBusMapper
ARMCortexM0PlusCT.l2_flusher	AsyncCacheFlushUnit

ARMCortexM0PlusCT contains the following CADI targets:

- ARM_Cortex-M0+

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.

Ports for ARMCortexM0PlusCT

Table 3-392: 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.

Name	Protocol	Type	Description
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

Initialize processor to big endian mode.

BKPT

Type

int

Default value

0x4

Number of breakpoint unit comparators implemented.

DBG**Type**

bool

Default value

0x1

Set whether debug extensions are implemented.

IOP**Type**

bool

Default value

0x0

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

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n].

NUM_IRQ**Type**

int

Default value

0x20

Number of user interrupts.

NUM_MPU_REGION**Type**

int

Default value

0x0

Number of MPU regions.

SYST**Type**

bool

Default value

0x1

Enable support for SysTick timer functionality.

USER**Type**

bool

Default value

0x0

Enable support for Unprivileged/Privileged Extension.

VTOR**Type**

bool

Default value

0x0

Include Vector Table Offset Register.

WIC**Type**

bool

Default value

0x1

Include support for WIC-mode deep sleep.

WPT**Type**

int

Default value

0x2

Number of watchpoint unit comparators implemented.

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base

Type
int

Default value
0x0

Virtual address of heap base.

semihosting-heap_limit

Type
int

Default value
0x20700000

Virtual address of top of heap.

semihosting-stack_base

Type
int

Default value
0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit

Type
int

Default value
0x20700000

Virtual address of stack limit.

3.5.52 ARMCortexM3CT

ARMCortexM3CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-393: IP revisions support

Revision	Quality level
r2p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM CortexM3CT

This model has the following Iris instances:

Table 3-394: ARM CortexM3CT Iris instances

InstanceName	ComponentName
ARM CortexM3CT	ARM_Cortex-M3
ARM CortexM3CT.acp_mapper	PVBusMapper
ARM CortexM3CT.ext_bus	PVBusLogger
ARM CortexM3CT.ext_bus.mapper	PVBusMapper
ARM CortexM3CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-395: ARM CortexM3CT MTI instances

InstanceName	ComponentName
ARM CortexM3CT	ARM_Cortex-M3
ARM CortexM3CT.acp_mapper	PVBusMapper
ARM CortexM3CT.ext_bus	PVBusLogger
ARM CortexM3CT.ext_bus.mapper	PVBusMapper
ARM CortexM3CT.l2_flusher	AsyncCacheFlushUnit

ARM CortexM3CT contains the following CADI targets:

- ARM_Cortex-M3

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.

Ports for ARM CortexM3CT

Table 3-397: 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.

Name	Protocol	Type	Description
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.
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 CortexM3CT

BB_PRESENT

Type

bool

Default value

0x1

Enable bitbanding.

BIGENDINIT

Type

bool

Default value

0x0

Initialize processor to big endian mode.

DBGLVL

Type

int

Default value

0x3

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

Number of bits of interrupt priority.

NUM_IRQ**Type**

int

Default value

0x10

Number of user interrupts.

NUM_MPU_REGION**Type**

int

Default value

0x8

Number of MPU regions.

TRACE_LVL**Type**

int

Default value

0x1

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

Include support for WIC-mode deep sleep.

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base

Type
int

Default value
0x0

Virtual address of heap base.

semihosting-heap_limit

Type
int

Default value
0x20700000

Virtual address of top of heap.

semihosting-stack_base

Type
int

Default value
0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit

Type
int

Default value
0x20700000

Virtual address of stack limit.

3.5.53 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 3-398: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM CortexM4CT

This model has the following Iris instances:

Table 3-399: ARM CortexM4CT Iris instances

InstanceName	ComponentName
ARM CortexM4CT	ARM_Cortex-M4
ARM CortexM4CT.acp_mapper	PVBusMapper
ARM CortexM4CT.ext_bus	PVBusLogger
ARM CortexM4CT.ext_bus.mapper	PVBusMapper
ARM CortexM4CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-400: ARM CortexM4CT MTI instances

InstanceName	ComponentName
ARM CortexM4CT	ARM_Cortex-M4
ARM CortexM4CT.acp_mapper	PVBusMapper
ARM CortexM4CT.ext_bus	PVBusLogger
ARM CortexM4CT.ext_bus.mapper	PVBusMapper
ARM CortexM4CT.l2_flusher	AsyncCacheFlushUnit

ARM CortexM4CT contains the following CADI targets:

- ARM_Cortex-M4

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 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.
- 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.
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.

Ports for ARMCortexM4CT

Table 3-402: 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.
dbgen	Signal	Slave	Disallow (DAP) debugger access.
edbgrq	Signal	Slave	External debug request.

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.
fpudisable	Signal	Slave	Configure core with no FPU on reset.
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).
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.
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 ARM CortexM4CT

BB_PRESENT

Type

bool

Default value

0x1

Enable bitbanding.

BIGENDINIT

Type

bool

Default value

0x0

Initialize processor to big endian mode.

DBGLVL**Type**

int

Default value

0x3

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

Number of bits of interrupt priority.

NUM_IRQ**Type**

int

Default value

0x10

Number of user interrupts.

NUM_MPU_REGION**Type**

int

Default value

0x8

Number of MPU regions.

TRACE_LVL**Type**

int

Default value

0x1

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

Include support for WIC-mode deep sleep.

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit**Type**

int

Default value

0x20700000

Virtual address of top of heap.

semihosting-stack_base**Type**

int

Default value

0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit**Type**

int

Default value

0x20700000

Virtual address of stack limit.

vfp-present**Type**

bool

Default value

0x1

Set whether the model has VFP support.

3.5.54 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 3-403: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexM7CT

This model has the following Iris instances:

Table 3-404: ARMCortexM7CT Iris 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

InstanceName	ComponentName
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 3-405: 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
ARMCortexM7CT.l1dcache	PVCache

InstanceName	ComponentName
ARMCortexM7CT.l1dcache.upstream[0]	PVBusSlave
ARMCortexM7CT.l1icache	PVCache
ARMCortexM7CT.l1icache.upstream[0]	PVBusSlave
ARMCortexM7CT.l2_flusher	AsyncCacheFlushUnit

ARMCortexM7CT contains the following CADI targets:

- ARM_Cortex-M7

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.

Ports for ARMCortexM7CT

Table 3-407: 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.

Name	Protocol	Type	Description
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 (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.
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.
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 ARM CortexM7CT

BIGENDINIT

Type

bool

Default value

0x0

Initialize processor to big endian mode.

CFG_AHBPSZ**Type**

int

Default value

0x0

Size of the AHBP port memory region. 0=AHBP disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

CTI**Type**

bool

Default value

0x0

CTI (Cross Trigger Interface) included.

CTI_irq0_pin**Type**

int

Default value

0x4

CTI interrupt request 0 pin.

CTI_irq1_pin**Type**

int

Default value

0x5

CTI interrupt request 1 pin.

DBGLVL**Type**

int

Default value

0x1

0: 2 DWT, 4 FPB; 1: 4 DWT, 8 FPB comparators.

DP_FLOAT

Type

bool

Default value

0x1

Support 8-byte floats.

INITAHBPEN

Type

bool

Default value

0x0

The AHBP enable state at reset.

INITVTOR

Type

int

Default value

0x0

vector-table offset at reset.

LVL_WIDTH

Type

int

Default value

0x3

Number of bits of interrupt priority.

NUM_IRQ

Type

int

Default value

0x20

Number of user interrupts.

NUM_MPU_REGION**Type**

int

Default value

0x10

Number of MPU regions.

TRC**Type**

bool

Default value

0x1

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

WIC**Type**

bool

Default value

0x1

Include support for WIC-mode deep sleep.

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

dcache-size**Type**

int

Default value

0x8000

L1 D-cache size in bytes.

dcache-state_modelled**Type**

bool

Default value

0x0

Set whether D-cache has stateful implementation.

dtcm_enable**Type**

bool

Default value

0x0

Enable DTCM at reset.

dtcm_size**Type**

int

Default value

0x100

DTCM size in KB.

icache-size**Type**

int

Default value

0x8000

L1 I-cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

itcm_enable**Type**

bool

Default value

0x0

Enable ITCM at reset.

itcm_size**Type**

int

Default value

0x100

ITCM size in KB.

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit**Type**

int

Default value

0x20700000

Virtual address of top of heap.

semihosting-stack_base**Type**

int

Default value

0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit**Type**

int

Default value

0x20700000

Virtual address of stack limit.

vfp-present**Type**

bool

Default value

0x1

Set whether the model has VFP support.

3.5.55 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 3-408: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexM23CT

This model has the following Iris instances:

Table 3-409: 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 3-410: 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

ARMCortexM23CT contains the following CADI targets:

- ARM_Cortex-M23

Differences between the CT model and RTL implementations

The model does not support MTB, ETM, or TPIU. MTB RAM is absent on the model.

Ports for ARMCortexM23CT

Table 3-411: 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.
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
dbgrestartd	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.

Name	Protocol	Type	Description
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.
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

Initialize processor to big endian mode.

BKPT**Type**

int

Default value

0x4

Number of breakpoint unit comparators implemented.

CTI**Type**

bool

Default value

0x0

CTI (Cross Trigger Interface) included.

CTI_irq0_pin**Type**

int

Default value

0x4

CTI interrupt request 0 pin.

CTI_irq1_pin**Type**

int

Default value

0x5

CTI interrupt request 1 pin.

DBG**Type**

bool

Default value

0x1

Set whether debug extensions are implemented.

INITVTOR**Type**

int

Default value

0x0

Secure vector-table offset at reset.

INITVTORNS**Type**

int

Default value

0x0

Non-Secure vector-table offset at reset.

IOP**Type**

bool

Default value

0x0

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

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

IRQDIS1**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

IRQDIS2**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

IRQDIS3**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

IRQDIS4**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

IRQDIS5**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

IRQDIS6**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

IRQDIS7**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

MPU_NS**Type**

int

Default value

0x8

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

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

NUMIRQ**Type**

int

Default value

0x10

Number of user interrupts.

SAU**Type**

int

Default value

0x4

Number of SAU regions (0 => no SAU).

SAU_CTRL.ALLNS**Type**

bool

Default value

0x0

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

Enable SAU at reset.

SAU_REGION0.BADDR**Type**

int

Default value

0x0

Base address of SAU region0 at reset.

SAU_REGION0.ENABLE**Type**

bool

Default value

0x0

Enable SAU region0 at reset.

SAU_REGION0.LADDR**Type**

int

Default value

0x0

Limit address of SAU region0 at reset.

SAU_REGION0.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region0 at reset.

SAU_REGION1.BADDR**Type**

int

Default value

0x0

Base address of SAU region1 at reset.

SAU_REGION1.ENABLE**Type**

bool

Default value

0x0

Enable SAU region1 at reset.

SAU_REGION1.LADDR**Type**

int

Default value

0x0

Limit address of SAU region1 at reset.

SAU_REGION1.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region1 at reset.

SAU_REGION2.BADDR**Type**

int

Default value

0x0

Base address of SAU region2 at reset.

SAU_REGION2.ENABLE**Type**

bool

Default value

0x0

Enable SAU region2 at reset.

SAU_REGION2.LADDR**Type**

int

Default value

0x0

Limit address of SAU region2 at reset.

SAU_REGION2.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region2 at reset.

SAU_REGION3.BADDR**Type**

int

Default value

0x0

Base address of SAU region3 at reset.

SAU_REGION3.ENABLE**Type**

bool

Default value

0x0

Enable SAU region3 at reset.

SAU_REGION3.LADDR**Type**

int

Default value

0x0

Limit address of SAU region3 at reset.

SAU_REGION3.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region3 at reset.

SAU_REGION4.BADDR**Type**

int

Default value

0x0

Base address of SAU region4 at reset.

SAU_REGION4.ENABLE**Type**

bool

Default value

0x0

Enable SAU region4 at reset.

SAU_REGION4.LADDR**Type**

int

Default value

0x0

Limit address of SAU region4 at reset.

SAU_REGION4.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region4 at reset.

SAU_REGION5.BADDR**Type**

int

Default value

0x0

Base address of SAU region5 at reset.

SAU_REGION5.ENABLE**Type**

bool

Default value

0x0

Enable SAU region5 at reset.

SAU_REGION5.LADDR**Type**

int

Default value

0x0

Limit address of SAU region5 at reset.

SAU_REGION5.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region5 at reset.

SAU_REGION6.BADDR**Type**

int

Default value

0x0

Base address of SAU region6 at reset.

SAU_REGION6.ENABLE**Type**

bool

Default value

0x0

Enable SAU region6 at reset.

SAU_REGION6.LADDR**Type**

int

Default value

0x0

Limit address of SAU region6 at reset.

SAU_REGION6.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region6 at reset.

SAU_REGION7.BADDR**Type**

int

Default value

0x0

Base address of SAU region7 at reset.

SAU_REGION7.ENABLE**Type**

bool

Default value

0x0

Enable SAU region7 at reset.

SAU_REGION7.LADDR**Type**

int

Default value

0x0

Limit address of SAU region7 at reset.

SAU_REGION7.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region7 at reset.

SECEXT**Type**

bool

Default value

0x1

Whether the ARMv8-M Security Extensions are included.

SYST**Type**

int

Default value

0x2

Include SysTick timer functionality (0=Absent, 1=Secure only, 2=Secure and NS).

VTOR**Type**

bool

Default value

0x1

Include Vector Table Offset Register.

WIC**Type**

bool

Default value

0x1

Include support for WIC-mode deep sleep.

WICLINES**Type**

int

Default value

0x12

Number of lines supported by the WIC interface.

WPT**Type**

int

Default value

0x4

Number of watchpoint unit comparators implemented.

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

has_core_dside_bus_gasket**Type**

bool

Default value

0x0

STL gasket enabled.

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit**Type**

int

Default value

0x20700000

Virtual address of top of heap.

semihosting-stack_base**Type**

int

Default value

0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit**Type**

int

Default value

0x20700000

Virtual address of stack limit.

3.5.56 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 3-412: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexM33CT

This model has the following Iris instances:

Table 3-413: 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 3-414: 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

ARMCortexM33CT contains the following CADI targets:

- ARM_Cortex-M33

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.

Ports for ARM CortexM33CT

Table 3-416: 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
dbgrestartd	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).

Name	Protocol	Type	Description
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.
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.
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.
wicenack	Signal	Master	Acknowledge signal for WICENREQ
wicenreq	Signal	Slave	Request for deep sleep to be WIC-based deep sleep.

Name	Protocol	Type	Description
wicsense[483]	Signal	Master	Indicates which input events can cause the WIC to generate the WAKEUP signal.

Parameters for ARM Cortex M33CT

BIGENDINIT

Type

bool

Default value

0x0

Initialize processor to big endian mode.

CDEMAPPEDONCP

Type

int

Default value

0xff

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

CDERTLID

Type

int

Default value

0x20

Value of ID_AFR0.CDERTLID.

CFGNOCECP

Type

int

Default value

0x0

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

CPIF

Type

bool

Default value

0x1

Specifies whether the external coprocessor interface is included.

CPNSPRESENT**Type**

int

Default value

0xff

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

CPSPPRESENT**Type**

int

Default value

0xff

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

CTI**Type**

bool

Default value

0x0

CTI (Cross Trigger Interface) included.

CTI_irq0_pin**Type**

int

Default value

0x4

CTI interrupt request 0 pin.

CTI_irq1_pin**Type**

int

Default value

0x5

CTI interrupt request 1 pin.

DBG_LVL

Type

int

Default value

0x2

0: Minimal debug; 1: 2 Watchpoints, 4 Breakpoint comparators; 2: 4 Watchpoints, 8 Breakpoint comparators.

DSP

Type

bool

Default value

0x1

Set whether the model has the DSP extension.

FPU

Type

bool

Default value

0x1

Set whether the model has VFP support.

INITNSVTOR

Type

int

Default value

0x0

Non-Secure vector-table offset at reset.

INITSVTOR

Type

int

Default value

0x0

Secure vector-table offset at reset.

IRQDIS0**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

IRQDIS1**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

IRQDIS10**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

IRQDIS11**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

IRQDIS12**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

IRQDIS13**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

IRQDIS14**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

IRQDIS2**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

IRQDIS3**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

IRQDIS4**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

IRQDIS5**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

IRQDIS6**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

IRQDIS7**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

IRQDIS8**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

IRQDIS9**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

IRQLVL**Type**

int

Default value

0x3

Number of bits of interrupt priority.

ITM**Type**

bool

Default value

0x1

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

LOCK_NS_MPU**Type**

bool

Default value

0x0

Lock down of Non-Secure MPU registers write.

LOCK_SAU**Type**

bool

Default value

0x0

Lock down of SAU registers write.

LOCK_S_MPU**Type**

bool

Default value

0x0

Lock down of Secure MPU registers write.

MPU_NS**Type**

int

Default value

0x8

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

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

NUMIRQ**Type**

int

Default value

0x20

Number of user interrupts.

SAU**Type**

int

Default value

0x4

Number of SAU regions (0 => no SAU).

SAU_CTRL.ALLNS**Type**

bool

Default value

0x0

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

Enable SAU at reset.

SAU_REGION0.BADDR**Type**

int

Default value

0x0

Base address of SAU region0 at reset.

SAU_REGION0.ENABLE**Type**

bool

Default value

0x0

Enable SAU region0 at reset.

SAU_REGION0.LADDR**Type**

int

Default value

0x0

Limit address of SAU region0 at reset.

SAU_REGION0.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region0 at reset.

SAU_REGION1.BADDR**Type**

int

Default value

0x0

Base address of SAU region1 at reset.

SAU_REGION1.ENABLE**Type**

bool

Default value

0x0

Enable SAU region1 at reset.

SAU_REGION1.LADDR**Type**

int

Default value

0x0

Limit address of SAU region1 at reset.

SAU_REGION1.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region1 at reset.

SAU_REGION2.BADDR**Type**

int

Default value

0x0

Base address of SAU region2 at reset.

SAU_REGION2.ENABLE**Type**

bool

Default value

0x0

Enable SAU region2 at reset.

SAU_REGION2.LADDR**Type**

int

Default value

0x0

Limit address of SAU region2 at reset.

SAU_REGION2.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region2 at reset.

SAU_REGION3.BADDR**Type**

int

Default value

0x0

Base address of SAU region3 at reset.

SAU_REGION3.ENABLE**Type**

bool

Default value

0x0

Enable SAU region3 at reset.

SAU_REGION3.LADDR**Type**

int

Default value

0x0

Limit address of SAU region3 at reset.

SAU_REGION3.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region3 at reset.

SAU_REGION4.BADDR**Type**

int

Default value

0x0

Base address of SAU region4 at reset.

SAU_REGION4.ENABLE**Type**

bool

Default value

0x0

Enable SAU region4 at reset.

SAU_REGION4.LADDR**Type**

int

Default value

0x0

Limit address of SAU region4 at reset.

SAU_REGION4.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region4 at reset.

SAU_REGION5.BADDR**Type**

int

Default value

0x0

Base address of SAU region5 at reset.

SAU_REGION5.ENABLE**Type**

bool

Default value

0x0

Enable SAU region5 at reset.

SAU_REGION5.LADDR**Type**

int

Default value

0x0

Limit address of SAU region5 at reset.

SAU_REGION5.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region5 at reset.

SAU_REGION6.BADDR**Type**

int

Default value

0x0

Base address of SAU region6 at reset.

SAU_REGION6.ENABLE**Type**

bool

Default value

0x0

Enable SAU region6 at reset.

SAU_REGION6.LADDR**Type**

int

Default value

0x0

Limit address of SAU region6 at reset.

SAU_REGION6.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region6 at reset.

SAU_REGION7.BADDR**Type**

int

Default value

0x0

Base address of SAU region7 at reset.

SAU_REGION7.ENABLE**Type**

bool

Default value

0x0

Enable SAU region7 at reset.

SAU_REGION7.LADDR**Type**

int

Default value

0x0

Limit address of SAU region7 at reset.

SAU_REGION7.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region7 at reset.

SECEXT**Type**

bool

Default value

0x1

Whether the ARMv8-M Security Extensions are included.

WIC**Type**

bool

Default value

0x1

Include support for WIC-mode deep sleep.

WICLINES**Type**

int

Default value

0x23

Number of lines supported by the WIC interface.

cde_impl_name**Type**

string

Default value

""

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

has_cde**Type**

bool

Default value

0x0

Enables Custom Datapath Extensions.

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit**Type**

int

Default value

0x20700000

Virtual address of top of heap.

semihosting-stack_base

Type
int

Default value
0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit

Type
int

Default value
0x20700000

Virtual address of stack limit.

vfp-enable_at_reset

Type
bool

Default value
0x0

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

3.5.57 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 3-417: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexM35PCT

This model has the following Iris instances:

Table 3-418: ARMCortexM35PCT Iris instances

InstanceName	ComponentName
ARMCortexM35PCT	ARM_Cortex-M35P

InstanceName	ComponentName
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 3-419: 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

ARMCortexM35PCT contains the following CADI targets:

- ARM_Cortex-M35P

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.

Field value	Description	Supported by model
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.

Ports for ARMCortexM35PCT

Table 3-421: 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
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.

Name	Protocol	Type	Description
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.
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

ATFCR is enabled when the core goes out of reset.

BIGENDINIT**Type**

bool

Default value

0x0

Initialize processor to big endian mode.

CPIF**Type**

bool

Default value

0x1

Specifies whether the external coprocessor interface is included.

CPNSPRESENT**Type**

int

Default value

0xff

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

CPSPPRESENT**Type**

int

Default value

0xff

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

CTI**Type**

bool

Default value

0x0

CTI (Cross Trigger Interface) included.

CTI_irq0_pin**Type**

int

Default value

0x4

CTI interrupt request 0 pin.

CTI_irq1_pin**Type**

int

Default value

0x5

CTI interrupt request 1 pin.

DBGLVL**Type**

int

Default value

0x2

0: Minimal debug; 1: 2 Watchpoints, 4 Breakpoint comparators; 2: 4 Watchpoints, 8 Breakpoint comparators.

DSP**Type**

bool

Default value

0x1

Set whether the model has the DSP extension.

FPU**Type**

bool

Default value

0x1

Set whether the model has VFP support.

INITNSVTOR**Type**

int

Default value

0x0

Non-Secure vector-table offset at reset.

INITSVTOR**Type**

int

Default value

0x0

Secure vector-table offset at reset.

IRQDIS0**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

IRQDIS1**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

IRQDIS10**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

IRQDIS11**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

IRQDIS12**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

IRQDIS13**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

IRQDIS14**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

IRQDIS2**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

IRQDIS3**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

IRQDIS4**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

IRQDIS5**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

IRQDIS6**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

IRQDIS7**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

IRQDIS8**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

IRQDIS9**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

IRQLVL**Type**

int

Default value

0x3

Number of bits of interrupt priority.

ITM**Type**

bool

Default value

0x1

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

LOCK_NS_MPU**Type**

bool

Default value

0x0

Lock down of Non-Secure MPU registers write.

LOCK_SAU**Type**

bool

Default value

0x0

Lock down of SAU registers write.

LOCK_S_MPU**Type**

bool

Default value

0x0

Lock down of Secure MPU registers write.

MPU_NS**Type**

int

Default value

0x8

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

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

NUMIRQ**Type**

int

Default value

0x20

Number of user interrupts.

SAU**Type**

int

Default value

0x4

Number of SAU regions (0 => no SAU).

SAU_CTRL.ALLNS**Type**

bool

Default value

0x0

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

Enable SAU at reset.

SAU_REGION0.BADDR**Type**

int

Default value

0x0

Base address of SAU region0 at reset.

SAU_REGION0.ENABLE**Type**

bool

Default value

0x0

Enable SAU region0 at reset.

SAU_REGION0.LADDR**Type**

int

Default value

0x0

Limit address of SAU region0 at reset.

SAU_REGION0.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region0 at reset.

SAU_REGION1.BADDR**Type**

int

Default value

0x0

Base address of SAU region1 at reset.

SAU_REGION1.ENABLE**Type**

bool

Default value

0x0

Enable SAU region1 at reset.

SAU_REGION1.LADDR**Type**

int

Default value

0x0

Limit address of SAU region1 at reset.

SAU_REGION1.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region1 at reset.

SAU_REGION2.BADDR**Type**

int

Default value

0x0

Base address of SAU region2 at reset.

SAU_REGION2.ENABLE**Type**

bool

Default value

0x0

Enable SAU region2 at reset.

SAU_REGION2.LADDR**Type**

int

Default value

0x0

Limit address of SAU region2 at reset.

SAU_REGION2.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region2 at reset.

SAU_REGION3.BADDR**Type**

int

Default value

0x0

Base address of SAU region3 at reset.

SAU_REGION3.ENABLE**Type**

bool

Default value

0x0

Enable SAU region3 at reset.

SAU_REGION3.LADDR**Type**

int

Default value

0x0

Limit address of SAU region3 at reset.

SAU_REGION3.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region3 at reset.

SAU_REGION4.BADDR**Type**

int

Default value

0x0

Base address of SAU region4 at reset.

SAU_REGION4.ENABLE**Type**

bool

Default value

0x0

Enable SAU region4 at reset.

SAU_REGION4.LADDR**Type**

int

Default value

0x0

Limit address of SAU region4 at reset.

SAU_REGION4.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region4 at reset.

SAU_REGION5.BADDR**Type**

int

Default value

0x0

Base address of SAU region5 at reset.

SAU_REGION5.ENABLE**Type**

bool

Default value

0x0

Enable SAU region5 at reset.

SAU_REGION5.LADDR**Type**

int

Default value

0x0

Limit address of SAU region5 at reset.

SAU_REGION5.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region5 at reset.

SAU_REGION6.BADDR**Type**

int

Default value

0x0

Base address of SAU region6 at reset.

SAU_REGION6.ENABLE**Type**

bool

Default value

0x0

Enable SAU region6 at reset.

SAU_REGION6.LADDR**Type**

int

Default value

0x0

Limit address of SAU region6 at reset.

SAU_REGION6.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region6 at reset.

SAU_REGION7.BADDR**Type**

int

Default value

0x0

Base address of SAU region7 at reset.

SAU_REGION7.ENABLE**Type**

bool

Default value

0x0

Enable SAU region7 at reset.

SAU_REGION7.LADDR**Type**

int

Default value

0x0

Limit address of SAU region7 at reset.

SAU_REGION7.NSC**Type**

bool

Default value

0x0

Set NSC for SAU region7 at reset.

SECEXT**Type**

bool

Default value

0x1

Whether the ARMv8-M Security Extensions are included.

WIC**Type**

bool

Default value

0x1

Include support for WIC-mode deep sleep.

WICLINES**Type**

int

Default value

0x23

Number of lines supported by the WIC interface.

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffffffff

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

0xffffffffffffffffffff

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit**Type**

int

Default value

0x20700000

Virtual address of top of heap.

semihosting-stack_base

Type
int

Default value
0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit

Type
int

Default value
0x20700000

Virtual address of stack limit.

vfp-enable_at_reset

Type
bool

Default value
0x0

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

3.5.58 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 3-422: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexM52CT

This model has the following Iris instances:

Table 3-423: 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 3-424: ARMCortexM52CT MTI instances

InstanceName	ComponentName
ARMCortexM52CT	ARM_Cortex-M52
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

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

ARMCortexM52CT contains the following CADI targets:

- ARM_Cortex-M52

About ARMCortexM52CT

This model supports the M-Profile Vector Extension (MVE) and the Custom Datapath Extension (CDE). For more information, see [CDE](#).

Ports for ARMCortexM52CT

Table 3-425: 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 (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

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 ARMCortexM52CT

CDEMAPPEDONCP

Type

int

Default value

0xff

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

CDERTLID**Type**

int

Default value

0x20

Value of ID_AFR0.CDERTLID.

CFGBIGEND**Type**

bool

Default value

0x0

Initialize processor to big endian mode.

CFGCPUINST**Type**

int

Default value

0x0

CPU instance number. This is part of the TCM base address, in bits 25:24.

CFGDTCMSZ**Type**

int

Default value

0x9

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

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

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.

CFGNOCDECP**Type**

int

Default value

0x0

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

CFGPACBTI**Type**

bool

Default value

0x0

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

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

Specifies whether the external coprocessor interface is included.

CPNSPRESENT**Type**

int

Default value

0xff

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

CPSPRESENT**Type**

int

Default value

0xff

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

CTI**Type**

bool

Default value

0x0

CTI (Cross Trigger Interface) included.

CTI_irq0_pin**Type**

int

Default value

0x4

CTI interrupt request 0 pin.

CTI_irq1_pin**Type**

int

Default value

0x5

CTI interrupt request 1 pin.

DBGLVL**Type**

int

Default value

0x2

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

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

DTCM Security Gate Unit included.

DTGUBLKSZ**Type**

int

Default value

0x3

DTCM gate unit block size. Size=pow(2, DTGUBLKSZ + 5) bytes.

DTGUMAXBLKS**Type**

int

Default value

0x0

Maximum number of DTCM gate unit blocks. Number of blocks=pow(2, DTGUMAXBLKS).

ECOREVNUM**Type**

int

Default value

0x0

ECO Revision number.

ERRDEVID.NUM**Type**

int

Default value

0x1

RAS: Number of implemented error record indexes, 0 to 1.

ETM**Type**

bool

Default value

0x1

Support for ETM trace. false : No ETM trace included, true: ETM trace included.

FPMVE**Type**

int

Default value

0x5

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

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

Non-Secure vector-table offset at reset.

INITPAHBEN**Type**

bool

Default value

0x0

The P-AHB enable state at reset.

INITSVTOR**Type**

int

Default value

0x0

Secure vector-table offset at reset.

INITTCMEN**Type**

int

Default value

0x3

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

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

IRQDIS1**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

IRQDIS10**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

IRQDIS11**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

IRQDIS12**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

IRQDIS13**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

IRQDIS14**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

IRQDIS2**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

IRQDIS3**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

IRQDIS4**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

IRQDIS5**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

IRQDIS6**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

IRQDIS7**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

IRQDIS8**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

IRQDIS9**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

IRQLVL**Type**

int

Default value

0x3

Number of bits of interrupt priority.

ITGU**Type**

bool

Default value

0x0

ITCM Security Gate Unit included.

ITGUBLKSZ**Type**

int

Default value

0x3

ITCM gate unit block size. Size= $\text{pow}(2, \text{ITGUBLKSZ} + 5)$ bytes.

ITGUMAXBLKS**Type**

int

Default value

0x0

Maximum number of ITCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{ITGUMAXBLKS})$.

ITM**Type**

bool

Default value

0x1

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

IWIC**Type**

bool

Default value

0x1

Include support for Internal Wake-up Interrupt Controller.

LOCKDTGU**Type**

bool

Default value

0x0

Lock down of Data TGU registers write.

LOCKITGU**Type**

bool

Default value

0x0

Lock down of Instruction TGU registers write.

LOCKTCM**Type**

bool

Default value

0x0

Lock down of TCM registers write.

LOCK_NS_MPU**Type**

bool

Default value

0x0

Lock down of Non-Secure MPU registers write.

LOCK_SAU**Type**

bool

Default value

0x0

Lock down of SAU registers write.

LOCK_S_MPU**Type**

bool

Default value

0x0

Lock down of Secure MPU registers write.

MPU_NS**Type**

int

Default value

0x8

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

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

NUMIRQ**Type**

int

Default value

0x20

Number of user interrupts.

SAU**Type**

int

Default value

0x4

Number of SAU regions (0 => no SAU).

SECEXT**Type**

bool

Default value

0x1

Whether the ARMv8-M Security Extensions are included.

UCACHE**Type**

bool

Default value

0x0

Whether the I-cache acts as a unified cache (ICACHESZ is used for the size).

WICLINES**Type**

int

Default value

0x23

Number of lines supported by the WIC interface.

cde_impl_name**Type**

string

Default value

""

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

dcache-state_modelled**Type**

bool

Default value

0x0

Set whether D-cache has stateful implementation.

delay_faultmask_update**Type**

bool

Default value

0x0

Delay FAULTMASK update to context sync.

delay_sysreg_update**Type**

bool

Default value

0x0

Delay some system register updates (e.g. SHCSR) to context sync.

ecc_on**Type**

bool

Default value

0x0

Enable Error Correcting Code.

has_cde**Type**

bool

Default value

0x0

Enables Custom Datapath Extensions.

has_core_dside_bus_gasket**Type**

bool

Default value

0x0

STL gasket enabled.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit**Type**

int

Default value

0x20700000

Virtual address of top of heap.

semihosting-stack_base**Type**

int

Default value

0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit**Type**

int

Default value

0x20700000

Virtual address of stack limit.

trace_style

Type
int

Default value
0x2

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

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

3.5.59 **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 3-426: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexM55CT

This model has the following Iris instances:

Table 3-427: ARMCortexM55CT Iris 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

InstanceName	ComponentName
ARMCortexM55CT.11_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[15]	PVBusSlave
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.11licache	PVCache
ARMCortexM55CT.11licache.upstream[0]	PVBusSlave
ARMCortexM55CT.12_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-428: ARMCortexM55CT MTI instances

InstanceName	ComponentName
ARMCortexM55CT	ARM_Cortex-M55
ARMCortexM55CT.acp_mapper	PVBusMapper
ARMCortexM55CT.ext_bus	PVBusLogger
ARMCortexM55CT.ext_bus.mapper	PVBusMapper
ARMCortexM55CT.11_incoherent_interconnect	PVCache
ARMCortexM55CT.11_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[17]	PVBusSlave

InstanceName	ComponentName
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.l1icache	PVCache
ARMCortexM55CT.l1icache.upstream[0]	PVBusSlave
ARMCortexM55CT.l2_flusher	AsyncCacheFlushUnit

ARMCortexM55CT contains the following CADI targets:

- ARM_Cortex-M55

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.
- Memory System Control Register (MSCR).
- Prefetcher Control Register (PFCR).
- Direct cache access registers.

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.

Ports for ARM CortexM55CT

Table 3-430: 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

Name	Protocol	Type	Description
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-M55-specific LOCKNSVTOR, LOCKSVTAIRCR, LOCKSMPU, LOCKNSMPU, LOCKSAU. Disable writes to VTOR_NS
lockpahb	Signal	Slave	P-AHB related ports Disable writes to PAHBRCR
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.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
prequest	PChannel	Slave	Low Power Interface

Name	Protocol	Type	Description
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.
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 ARM CortexM55CT

CDEMAPPEDONCP

Type

int

Default value

0xff

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

CDERTLID

Type

int

Default value

0x20

Value of ID_AFR0.CDERTLID.

CFGBIGEND

Type

bool

Default value

0x0

Initialize processor to big endian mode.

CFGDTCMSZ**Type**

int

Default value

0x9

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

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

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

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

CFGPAHBSZ**Type**

int

Default value

0x0

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

Specifies whether the external coprocessor interface is included.

CPNSPRESENT**Type**

int

Default value

0xff

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

CPSPRESENT**Type**

int

Default value

0xff

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

CTI**Type**

bool

Default value

0x0

CTI (Cross Trigger Interface) included.

CTI_irq0_pin**Type**

int

Default value

0x4

CTI interrupt request 0 pin.

CTI_irq1_pin**Type**

int

Default value

0x5

CTI interrupt request 1 pin.

DBGLVL**Type**

int

Default value

0x2

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

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

DTCM Security Gate Unit included.

DTGUBLKSZ**Type**

int

Default value

0x3

DTCM gate unit block size. Size= $\text{pow}(2, \text{DTGUBLKSZ} + 5)$ bytes.

DTGUMAXBLKS**Type**

int

Default value

0x0

Maximum number of DTCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{DTGUMAXBLKS})$.

ECOREVNUM**Type**

int

Default value

0x0

ECO Revision number.

ERRDEVID . NUM**Type**

int

Default value

0x1

RAS: Number of implemented error record indexes, 0 to 1.

ETM**Type**

bool

Default value

0x1

Support for ETM trace. false : No ETM trace included, true: ETM trace included.

FPU**Type**

bool

Default value

0x1

Set whether the model has VFP support.

ICACHESZ**Type**

int

Default value

0xf

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

Non-Secure vector-table offset at reset.

INITPAHBEN**Type**

bool

Default value

0x0

The P-AHB enable state at reset.

INITSVTOR**Type**

int

Default value

0x0

Secure vector-table offset at reset.

INITTCMEN**Type**

int

Default value

0x3

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

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

IRQDIS1**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

IRQDIS10**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

IRQDIS11**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

IRQDIS12**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

IRQDIS13**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

IRQDIS14**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

IRQDIS2**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

IRQDIS3**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

IRQDIS4**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

IRQDIS5**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

IRQDIS6**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

IRQDIS7**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

IRQDIS8**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

IRQDIS9**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

IRQLVL**Type**

int

Default value

0x3

Number of bits of interrupt priority.

ITGU**Type**

bool

Default value

0x0

ITCM Security Gate Unit included.

ITGUBLKSZ**Type**

int

Default value

0x3

ITCM gate unit block size. Size= $\text{pow}(2, \text{ITGUBLKSZ} + 5)$ bytes.

ITGUMAXBLKS**Type**

int

Default value

0x0

Maximum number of ITCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{ITGUMAXBLKS})$.

ITM**Type**

bool

Default value

0x1

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

IWIC**Type**

bool

Default value

0x1

Include support for Internal Wake-up Interrupt Controller.

LOCKDTGU**Type**

bool

Default value

0x0

Lock down of Data TGU registers write.

LOCKITGU**Type**

bool

Default value

0x0

Lock down of Instruction TGU registers write.

LOCKTCM**Type**

bool

Default value

0x0

Lock down of TCM registers write.

LOCK_NS_MPU**Type**

bool

Default value

0x0

Lock down of Non-Secure MPU registers write.

LOCK_SAU**Type**

bool

Default value

0x0

Lock down of SAU registers write.

LOCK_S_MPU**Type**

bool

Default value

0x0

Lock down of Secure MPU registers write.

MPU_NS**Type**

int

Default value

0x8

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

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

MVE**Type**

int

Default value

0x1

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

Number of user interrupts.

SAU**Type**

int

Default value

0x4

Number of SAU regions (0 => no SAU).

SECEXT**Type**

bool

Default value

0x1

Whether the ARMv8-M Security Extensions are included.

WICLINES**Type**

int

Default value

0x23

Number of lines supported by the WIC interface.

cde_impl_name**Type**

string

Default value

""

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

dcache-state_modelled**Type**

bool

Default value

0x0

Set whether D-cache has stateful implementation.

delay_faultmask_update**Type**

bool

Default value

0x0

Delay FAULTMASK update to context sync.

delay_sysreg_update**Type**

bool

Default value

0x0

Delay some system register updates (e.g. SHCSR) to context sync.

ecc_on**Type**

bool

Default value

0x0

Enable Error Correcting Code.

has_cde**Type**

bool

Default value

0x0

Enables Custom Datapath Extensions.

has_core_dside_bus_gasket**Type**

bool

Default value

0x0

STL gasket enabled.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit**Type**

int

Default value

0x20700000

Virtual address of top of heap.

semihosting-stack_base**Type**

int

Default value

0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit

Type
int

Default value
0x20700000

Virtual address of stack limit.

trace_style

Type
int

Default value
0x2

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

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

3.5.60 **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 3-431: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexM85CT

This model has the following Iris instances:

Table 3-432: ARMCortexM85CT Iris instances

InstanceName	ComponentName
ARMCortexM85CT	ARM_Cortex-M85
ARMCortexM85CT.acp_mapper	PVBusMapper
ARMCortexM85CT.ext_bus	PVBusLogger
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

This model has the following MTI trace components:

Table 3-433: ARMCortexM85CT MTI instances

InstanceName	ComponentName
ARMCortexM85CT	ARM_Cortex-M85
ARMCortexM85CT.acp_mapper	PVBusMapper
ARMCortexM85CT.ext_bus	PVBusLogger
ARMCortexM85CT.ext_bus.mapper	PVBusMapper
ARMCortexM85CT.l1_incoherent_interconnect	PVCache
ARMCortexM85CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexM85CT.11_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexM85CT.11_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexM85CT.11dcache	PVCache
ARMCortexM85CT.11dcache.upstream[0]	PVBusSlave
ARMCortexM85CT.11icache	PVCache
ARMCortexM85CT.11icache.upstream[0]	PVBusSlave
ARMCortexM85CT.12_flusher	AsyncnCacheFlushUnit

ARMCortexM85CT contains the following CADI targets:

- ARM_Cortex-M85

Ports for ARMCortexM85CT

Table 3-434: 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	-

Name	Protocol	Type	Description
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 PAHBCR
locksau	Signal	Slave	Disable writes to the SAU_* registers
locksmpu	Signal	Slave	Disable writes to the Secure MPU_* registers

Name	Protocol	Type	Description
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.
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.
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 CortexM85CT

CDEMAPPEDONCP

Type

int

Default value

0xff

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

CDERTLID

Type

int

Default value

0x20

Value of ID_AFR0.CDERTLID.

CFGBIGEND**Type**

bool

Default value

0x0

Initialize processor to big endian mode.

CFGCPUINST**Type**

int

Default value

0x0

CPU instance number. This is part of the TCM base address, in bits 25:24.

CFGDTCMSZ**Type**

int

Default value

0x9

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

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

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.

CFGNOCDCECP**Type**

int

Default value

0x0

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

CFGPAACBTI**Type**

bool

Default value

0x0

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

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

Specifies whether the external coprocessor interface is included.

CPNSPRESENT**Type**

int

Default value`0xff`

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

CPSPRESENT**Type**`int`**Default value**`0xff`

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

CTI**Type**`bool`**Default value**`0x0`

CTI (Cross Trigger Interface) included.

CTI_irq0_pin**Type**`int`**Default value**`0x4`

CTI interrupt request 0 pin.

CTI_irq1_pin**Type**`int`**Default value**`0x5`

CTI interrupt request 1 pin.

DBGLVL**Type**`int`**Default value**`0x2`

1: 4 Watchpoints, 4 Breakpoint comparators; 2: 8 Watchpoints, 8 Breakpoint comparators.

DCACHESZ

Type

int

Default value

0xf

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

DTCM Security Gate Unit included.

DTGUBLKSZ

Type

int

Default value

0x3

DTCM gate unit block size. Size=pow(2, DTGUBLKSZ + 5) bytes.

DTGUMAXBLKS

Type

int

Default value

0x0

Maximum number of DTCM gate unit blocks. Number of blocks=pow(2, DTGUMAXBLKS).

ECOREVNUM

Type

int

Default value

0x0

ECO Revision number.

ERRDEVID.NUM**Type**

int

Default value

0x1

RAS: Number of implemented error record indexes, 0 to 1.

ETM**Type**

bool

Default value

0x1

Support for ETM trace. false : No ETM trace included, true: ETM trace included.

FPU**Type**

bool

Default value

0x1

Set whether the model has VFP support.

ICACHESZ**Type**

int

Default value

0xf

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

Non-Secure vector-table offset at reset.

INITPAHBEN**Type**

bool

Default value

0x0

The P-AHB enable state at reset.

INITSVTOR**Type**

int

Default value

0x0

Secure vector-table offset at reset.

INITTCMEN**Type**

int

Default value

0x3

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

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

IRQDIS1**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

IRQDIS10**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

IRQDIS11**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

IRQDIS12**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

IRQDIS13**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

IRQDIS14**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

IRQDIS2**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

IRQDIS3**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

IRQDIS4**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

IRQDIS5**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

IRQDIS6**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

IRQDIS7**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

IRQDIS8**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

IRQDIS9**Type**

int

Default value

0x0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

IRQLVL**Type**

int

Default value

0x3

Number of bits of interrupt priority.

ITGU**Type**

bool

Default value

0x0

ITCM Security Gate Unit included.

ITGUBLKSZ**Type**

int

Default value

0x3

ITCM gate unit block size. Size= $\text{pow}(2, \text{ITGUBLKSZ} + 5)$ bytes.

ITGUMAXBLKS**Type**

int

Default value

0x0

Maximum number of ITCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{ITGUMAXBLKS})$.

ITM**Type**

bool

Default value

0x1

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

IWIC**Type**

bool

Default value

0x1

Include support for Internal Wake-up Interrupt Controller.

LOCKDTGU**Type**

bool

Default value

0x0

Lock down of Data TGU registers write.

LOCKITGU**Type**

bool

Default value

0x0

Lock down of Instruction TGU registers write.

LOCKTCM**Type**

bool

Default value

0x0

Lock down of TCM registers write.

LOCK_NS_MPU**Type**

bool

Default value

0x0

Lock down of Non-Secure MPU registers write.

LOCK_SAU**Type**

bool

Default value

0x0

Lock down of SAU registers write.

LOCK_S_MPU**Type**

bool

Default value

0x0

Lock down of Secure MPU registers write.

MPU_NS**Type**

int

Default value

0x8

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

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

MVE**Type**

int

Default value

0x1

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

Number of user interrupts.

SAU**Type**

int

Default value

0x4

Number of SAU regions (0 => no SAU).

SECEXT**Type**

bool

Default value

0x1

Whether the ARMv8-M Security Extensions are included.

WICLINES**Type**

int

Default value

0x23

Number of lines supported by the WIC interface.

cde_impl_name**Type**

string

Default value

""

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

dcache-state_modelled**Type**

bool

Default value

0x0

Set whether D-cache has stateful implementation.

delay_faultmask_update**Type**

bool

Default value

0x0

Delay FAULTMASK update to context sync.

delay_sysreg_update**Type**

bool

Default value

0x0

Delay some system register updates (e.g. SHCSR) to context sync.

ecc_on**Type**

bool

Default value

0x0

Enable Error Correcting Code.

has_cde**Type**

bool

Default value

0x0

Enables Custom Datapath Extensions.

has_core_dside_bus_gasket**Type**

bool

Default value

0x0

STL gasket enabled.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit**Type**

int

Default value

0x20700000

Virtual address of top of heap.

semihosting-stack_base**Type**

int

Default value

0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit**Type**

int

Default value

0x20700000

Virtual address of stack limit.

trace_style**Type**

int

Default value

0x2

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

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

3.5.61 ARM CortexR4CT

ARM CortexR4CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-435: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM CortexR4CT

This model has the following Iris instances:

Table 3-436: ARM CortexR4CT Iris instances

InstanceName	ComponentName
ARM CortexR4CT	ARM_Cortex-R4
ARM CortexR4CT.acp_mapper	PVBusMapper
ARM CortexR4CT.cpu0.l1dcache	PVCache
ARM CortexR4CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM CortexR4CT.cpu0.l1icache	PVCache
ARM CortexR4CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARM CortexR4CT.ext_bus	PVBusLogger
ARM CortexR4CT.ext_bus.mapper	PVBusMapper
ARM CortexR4CT.l1_incoherent_interconnect	PVCache
ARM CortexR4CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARM CortexR4CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARM CortexR4CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARM CortexR4CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARM CortexR4CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARM CortexR4CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARM CortexR4CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave

InstanceName	ComponentName
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

This model has the following MTI trace components:

Table 3-437: 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.l1icache	PVCache
ARMCortexR4CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR4CT.ext_bus	PVBusLogger
ARMCortexR4CT.ext_bus.mapper	PVBusMapper
ARMCortexR4CT.11_incoherent_interconnect	PVCache
ARMCortexR4CT.11_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[10]	PVBusSlave
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

InstanceName	ComponentName
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

ARMCortexR4CT contains the following CADI targets:

- ARM_Cortex-R4
- PVMCache

About ARMCortexR4CT

- The model implements the `cfgie` port, although it is optional in hardware.
- `pmbus_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.

Ports for ARMCortexR4CT

Table 3-438: Ports

Name	Protocol	Type	Description
<code>cfgend0</code>	Signal	Slave	Configure BE8 mode after a reset.
<code>cfgie</code>	Signal	Slave	Configure big endian instruction format after a reset.
<code>cfgnmfi</code>	Signal	Slave	Configure FIQs as non-maskable after a reset.
<code>cfgte</code>	Signal	Slave	Configure exceptions to be taken in thumb mode after a reset.
<code>clk_in</code>	ClockSignal	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the core executes instructions.
<code>cpuhalt</code>	Signal	Slave	Raising this signal will put the core into halt mode.
<code>fiq</code>	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
<code>initramd</code>	Signal	Slave	Configure DTCM enabled after a reset.
<code>initrami</code>	Signal	Slave	Configure ITCM enabled after a reset.
<code>irq</code>	Signal	Slave	This signal drives the CPU's interrupt handling.
<code>loczrama</code>	Signal	Slave	Location of ATCM at reset.
<code>pmuirq</code>	Signal	Master	Interrupt signal from performance monitoring unit.
<code>pvbus_m</code>	PVBus	Master	The core will generate bus requests on this port.
<code>pvbus_s</code>	PVBus	Slave	Slave access to TCMs.
<code>reset</code>	Signal	Slave	Raising this signal will put the core into reset mode.
<code>standbywfi</code>	Signal	Master	Signal from the core that it is waiting in standby for an interrupt.

Name	Protocol	Type	Description
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.
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 CortexR4CT

CFGEND0

Type

bool

Default value

0x0

Initialize to BE8 endianness.

CFGIE

Type

bool

Default value

0x0

Set the reset value of the instruction endian bit.

CFGNMF1

Type

bool

Default value

0x0

Enable nonmaskable FIQ interrupts on startup.

CFGTE

Type

bool

Default value

0x0

Initialize to take exceptions in T32 state. Model starts in T32 state.

INITRAMD**Type**

bool

Default value

0x0

Set or reset the INITRAMD signal.

INITRAMI**Type**

bool

Default value

0x0

Set or reset the INITRAMI signal.

LOCZRAMI**Type**

bool

Default value

0x0

Set or reset the LOCZRAMI signal.

NUM_MPU_REGION**Type**

int

Default value

0x8

Number of MPU regions.

VINITHI**Type**

bool

Default value

0x0

Initialize with high vectors enabled.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

dcache-size**Type**

int

Default value

0x10000

Set D-cache size in bytes.

dcache-state_modelled**Type**

bool

Default value

0x0

Set whether D-cache has stateful implementation.

dtcm0_base**Type**

int

Default value

0x800000

Base address of DTCM at startup.

dtcm0_size**Type**

int

Default value

0x8

Size of DTCM in KB.

icache-size**Type**

int

Default value

0x10000

Set I-cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

implements_vfp**Type**

bool

Default value

0x1

Set whether the model has been built with VFP support.

itcm0_base**Type**

int

Default value

0x0

Base address of ITCM at startup.

itcm0_size**Type**

int

Default value

0x8

Size of ITCM in KB.

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

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

ARM HLT number for semihosting.

semihosting-ARM_SVC**Type**

int

Default value

0x123456

ARM SVC number for semihosting.

semihosting-Thumb_HLT**Type**

int

Default value

0x3c

Thumb HLT number for semihosting.

semihosting-Thumb_SVC**Type**

int

Default value

0xab

Thumb SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

semihosting-hlt-enable**Type**

bool

Default value

0x0

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

Prefix semihosting output with target instance name.

semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

vfp-enable_at_reset**Type**

bool

Default value

0x0

Enable coprocessor access and VFP at reset.

3.5.62 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 3-439: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexR5x1CT

This model has the following Iris instances:

Table 3-440: 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
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

InstanceName	ComponentName
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 3-441: 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
ARMCortexR5x1CT.l2_flusher	AsyncCacheFlushUnit

ARMCortexR5x1CT contains the following CADI targets:

- ARM_Cortex-R5

- Cluster_ARM_Cortex-R5

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.

pvb_{us}_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 SCTL_R 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:

- The software enables the VIC interface by setting the VE bit in the CP15 control register and setting up suitable interrupt routines.
- The VIC asserts IRQ.
- Some time later, the processor detects and responds to the IRQ by asserting `vic_ack`.
- The VIC writes the vector address to the processor using `vic_addr`.
- The processor de-asserts `vic_ack`.
- 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.

Ports for ARMCortexR5x1CT

Table 3-442: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	ACP slave port.
<code>cfgatcmsz[2]</code>	Value	Slave	ATCM size.
<code>cfgbtcmsz[2]</code>	Value	Slave	BTCLM Size.
<code>cfgend[2]</code>	Signal	Slave	This signal is for EE bit initialisation. This is CFGEE in RTL but <code>cfgend</code> here fastsim consistency reasons.
<code>cfgnmfi[2]</code>	Signal	Slave	Controls non-maskable Fast Interrupts.
<code>clk_in</code>	ClockSignal	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the core executes instructions.
<code>cpuhalt[2]</code>	Signal	Slave	Raising this signal will put the core into halt mode.
<code>event[2]</code>	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
<code>fiq[2]</code>	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
<code>groupid</code>	Value	Slave	Group ID used for MPIDR.
<code>initrama[2]</code>	Signal	Slave	If ATCM is enabled at reset.
<code>initramb[2]</code>	Signal	Slave	If BTCLM is enabled at reset.

Name	Protocol	Type	Description
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

Value read in GROUP ID register field, bits[15:8] of the MPIDR.

INST_ENDIAN

Type

bool

Default value

0x1

Controls whether the model supports the instruction endianness bit.

LOCK_STEP

Type

int

Default value

0x0

Affects dual-processor configurations only, and ignored by single-processor configurations.

MICRO_SCU**Type**

bool

Default value

0x1

Controls whether the effects of the MicroSCU are modeled.

NUM_BREAKPOINTS**Type**

int

Default value

0x3

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

Sets the number of MPU regions.

NUM_WATCHPOINTS**Type**

int

Default value

0x2

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

Sets whether the model starts in split mode or locked mode.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGATCMSZ**Type**

int

Default value

0xe

Sets the size of the ATCM.

cpuX.CFGBTCMSZ**Type**

int

Default value

0xe

Sets the size of the BTCM.

cpuX.CFGEND**Type**

bool

Default value

0x0

Initialize to BE8 endianness.

cpuX.CFGIE**Type**

bool

Default value

0x0

Set the reset value of the instruction endian bit.

cpuX.CFGNMFI**Type**

bool

Default value

0x0

Enable nonmaskable FIQ interrupts on startup.

cpuX.DP_FLOAT**Type**

bool

Default value

0x1

Sets whether double-precision instructions are available.

cpuX.INITRAMA**Type**

bool

Default value

0x0

Initialize with TCMA enabled.

cpuX.INITRAMB**Type**

bool

Default value

0x0

Initialize with TCMB enabled.

cpuX.LOCZRAMA**Type**

bool

Default value

0x0

Initialize with LOCZRAMA set to 1.

cpuX.TEINIT**Type**

bool

Default value

0x0

T32 exception enable. The default has exceptions including reset handled in A32 state.

cpuX.VINITHI**Type**

bool

Default value

0x0

Initialize with high vectors enabled.

cpuX.atcm_base**Type**

int

Default value

0x40000000

Model-specific. Sets the base address of the ATCM (forced to 0 if LOCZRAMA is 1).

cpuX.btcn_base**Type**

int

Default value

0x0

Model-specific. Sets the base address of the BTCM (forced to 0 if LOCZRAMA is 0).

cpuX.dcache-size**Type**

int

Default value

0x10000

Set D-cache size in bytes.

cpuX.icache-size**Type**

int

Default value

0x10000

Set I-cache size in bytes.

cpuX.min_sync_level**Type**

int

Default value

0x0

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-ARM_HLT**Type**

int

Default value

0xf000

ARM HLT number for semihosting.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

ARM SVC number for semihosting.

cpuX.semihosting-Thumb_HLT**Type**

int

Default value

0x3c

Thumb HLT number for semihosting.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

Thumb SVC number for semihosting.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

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

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-hlt-enable**Type**

bool

Default value

0x0

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

Prefix semihosting output with target instance name.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

Enable coprocessor access and VFP at reset.

cpuX.vfp-present**Type**

bool

Default value

0x1

Set whether model has VFP support.

dcache-state_modelled**Type**

bool

Default value

0x0

Set whether D-cache has stateful implementation.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

reported_patch_level

Type
int

Default value
0xffffffffffffffff

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

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

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!.

3.5.63 **ARMCortexR7x1CT**

ARMCortexR7x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-443: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexR7x1CT

This model has the following Iris instances:

Table 3-444: ARMCortexR7x1CT Iris instances

InstanceName	ComponentName
ARMCortexR7x1CT	Cluster_ARM_Cortex-R7
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

This model has the following MTI trace components:

Table 3-445: 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

InstanceName	ComponentName
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

ARMCortexR7x1CT contains the following CADI targets:

- ARM_Cortex-R7
- Cluster_ARM_Cortex-R7
- PVCache

About ARMCortexR7x1CT

An ARMCortexR7x2CT component also exists.

The per-core parameters are preceded by cpun., where n identifies the core (0 or 1).

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

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.

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:

- 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.
- ECC and parity schemes are hardware-specific so are not supported.

Ports for ARMCortexR7x1CT

Table 3-446: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>cfgend[1]</code>	Signal	Slave	This signal if 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.
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.
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 ARMCortexR7x1CT

CLUSTER_ID

Type

int

Default value

0x0

Processor cluster ID value.

LOCK_STEP

Type

int

Default value

0x0

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

Enables filtering of address ranges.

MFILTEREND

Type

int

Default value

0x0

Specifies the end address for address filtering.

MFILTERSTART

Type

int

Default value

0x0

Specifies the start address for address filtering.

NUM_MPU_REGION**Type**

int

Default value

0xc

Sets the number of MPU regions.

PERIPHBASE**Type**

int

Default value

0xae000000

Base address of peripheral memory space.

PFILTEREND**Type**

int

Default value

0x0

Specifies the end address for peripheral port address filtering.

PFILTERSTART**Type**

int

Default value

0xffff0000

Specifies the start address for peripheral port address filtering.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

`cpi_mul`**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

`cpuX.CFGEND`**Type**

bool

Default value

0x0

Initialize to BE8 endianness.

`cpuX.CFGNMFI`**Type**

bool

Default value

0x0

Enable nonmaskable FIQ interrupts on startup.

`cpuX.DP_FLOAT`**Type**

bool

Default value

0x1

Sets whether double-precision instructions are available.

`cpuX.INITRAM`**Type**

bool

Default value

0x0

Enable the processor to boot from the instruction TCM.

cpuX . POWERCTLI**Type**

int

Default value

0x0

Default power control state for processor.

cpuX . SMPnAMP**Type**

bool

Default value

0x0

Set whether the processor is part of a coherent domain.

cpuX . TEINIT**Type**

bool

Default value

0x0

T32 exception enable. The default has exceptions including reset handled in A32 state.

cpuX . VINITHI**Type**

bool

Default value

0x0

Initialize with high vectors enabled.

cpuX . dcache-size**Type**

int

Default value

0x8000

Set D-cache size in bytes.

cpuX.dtcn_size**Type**

int

Default value

0x8

DTCM size in KB.

cpuX.icache-size**Type**

int

Default value

0x8000

Set I-cache size in bytes.

cpuX.itcm_size**Type**

int

Default value

0x8

ITCM size in KB.

cpuX.min_sync_level**Type**

int

Default value

0x0

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-ARM_HLT**Type**

int

Default value

0xf000

ARM HLT number for semihosting.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

ARM SVC number for semihosting.

cpuX.semihosting-Thumb_HLT**Type**

int

Default value

0x3c

Thumb HLT number for semihosting.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

Thumb SVC number for semihosting.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

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

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-hlt-enable**Type**

bool

Default value

0x0

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

Prefix semihosting output with target instance name.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.tcm-present**Type**

bool

Default value

0x1

Disables the DTCM and ITCM.

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

Enable coprocessor access and VFP at reset.

cpuX.vfp-present**Type**

bool

Default value

0x1

Set whether model has VFP support.

dcache-state_modelled**Type**

bool

Default value

0x0

Set whether D-cache has stateful implementation.

dic-spi_count**Type**

int

Default value

0x40

Number of shared peripheral interrupts implemented.

ecc_on**Type**

bool

Default value

0x0

Enable Error Correcting Code.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

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 **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 3-447: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexR8x1CT

This model has the following Iris instances:

Table 3-448: ARMCortexR8x1CT Iris instances

InstanceName	ComponentName
ARMCortexR8x1CT	Cluster_ARM_Cortex-R8
ARMCortexR8x1CT.acp_mapper	PVBusMapper
ARMCortexR8x1CT.cpu0	ARM_Cortex-R8
ARMCortexR8x1CT.cpu0.l1dcache	PVCache
ARMCortexR8x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR8x1CT.cpu0.l1icache	PVCache

InstanceName	ComponentName
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 3-449: ARMCortexR8x1CT MTI instances

InstanceName	ComponentName
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

InstanceName	ComponentName
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexR8x1CT.11_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexR8x1CT.12_flusher	AsyncCacheFlushUnit

ARMCortexR8x1CT contains the following CADI targets:

- ARM_Cortex-R8
- Cluster_ARM_Cortex-R8
- PVCache

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.

Ports for ARMCortexR8x1CT

Table 3-450: 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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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

Processor cluster ID value.

FPFILTEREND0

Type

int

Default value

0x0

Specifies the end address for the fast peripheral port address filtering.

FPFILTEREND1

Type

int

Default value

0x0

Specifies the end address for the fast peripheral port address filtering.

FPFILTEREND2

Type

int

Default value

0x0

Specifies the end address for the fast peripheral port address filtering.

FPFILTEREND3**Type**

int

Default value

0x0

Specifies the end address for the fast peripheral port address filtering.

FPFILTERSTART0**Type**

int

Default value

0xffff00000

Specifies the start address for the fast peripheral port address filtering.

FPFILTERSTART1**Type**

int

Default value

0xffff00000

Specifies the start address for the fast peripheral port address filtering.

FPFILTERSTART2**Type**

int

Default value

0xffff00000

Specifies the start address for the fast peripheral port address filtering.

FPFILTERSTART3**Type**

int

Default value

0xffff00000

Specifies the start address for the fast peripheral port address filtering.

LOCK_STEP**Type**

int

Default value

0x0

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

Enables filtering of address ranges.

MFILTEREND**Type**

int

Default value

0x0

Specifies the end address for address filtering.

MFILTERSTART**Type**

int

Default value

0x0

Specifies the start address for address filtering.

NUM_MPU_REGION**Type**

int

Default value

0xc

Sets the number of MPU regions.

PERIPHBASE**Type**

int

Default value

0xae000000

Base address of peripheral memory space.

PFILTEREND**Type**

int

Default value

0x0

Specifies the end address for peripheral port address filtering.

PFILTERSTART**Type**

int

Default value

0xffff0000

Specifies the start address for peripheral port address filtering.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Initialize to BE8 endianness.

cpuX.CFGNMFI**Type**

bool

Default value

0x0

Enable nonmaskable FIQ interrupts on startup.

cpuX.DP_FLOAT**Type**

bool

Default value

0x1

Sets whether double-precision instructions are available.

cpuX.INITRAM**Type**

bool

Default value

0x0

Enable the processor to boot from the instruction TCM.

cpuX.POWERCTLI**Type**

int

Default value

0x0

Default power control state for processor.

cpuX.SMPnAMP**Type**

bool

Default value

0x0

Set whether the processor is part of a coherent domain.

cpuX.TEINIT**Type**

bool

Default value

0x0

T32 exception enable. The default has exceptions including reset handled in A32 state.

cpuX.VINITHI**Type**

bool

Default value

0x0

Initialize with high vectors enabled.

cpuX.dcache-size**Type**

int

Default value

0x8000

Set D-cache size in bytes.

cpuX.dtcn_size**Type**

int

Default value

0x8

DTCM size in KB.

cpuX.icache-size**Type**

int

Default value

0x8000

Set I-cache size in bytes.

cpuX.itcm_size**Type**

int

Default value

0x8

ITCM size in KB.

cpuX.min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-ARM_HLT**Type**

int

Default value

0xf000

ARM HLT number for semihosting.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

ARM SVC number for semihosting.

cpuX.semihosting-Thumb_HLT**Type**

int

Default value

0x3c

Thumb HLT number for semihosting.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

Thumb SVC number for semihosting.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

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

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-hlt-enable**Type**

bool

Default value

0x0

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

Prefix semihosting output with target instance name.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.tcm-present**Type**

bool

Default value

0x1

Disables the DTCM and ITCM.

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

Enable coprocessor access and VFP at reset.

cpuX.vfp-present**Type**

bool

Default value

0x1

Set whether model has VFP support.

dcache-state_modelled**Type**

bool

Default value

0x0

Set whether D-cache has stateful implementation.

dic-spi_count**Type**

int

Default value

0x40

Number of shared peripheral interrupts implemented.

ecc_on**Type**

bool

Default value

0x0

Enable Error Correcting Code.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

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 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 3-451: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Changes in 11.27.17

Parameters added:

- has_llpp

Iris and MTI instances for ARMCortexR52PlusCT

This model has the following Iris instances:

Table 3-452: 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

InstanceName	ComponentName
ARMCortexR52PlusCT.cpu0	ARM_Cortex-R52Plus
ARMCortexR52PlusCT.cpu0.UTLB	TLB
ARMCortexR52PlusCT.cpu0.dtlb	TlbCadi
ARMCortexR52PlusCT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR52PlusCT.cpu0.l1dcache	PVCache
ARMCortexR52PlusCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR52PlusCT.cpu0.l1licache	PVCache
ARMCortexR52PlusCT.cpu0.l1licache.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
ARMCortexR52PlusCT.gic_iri.rd_t1	GICv3Distributor
ARMCortexR52PlusCT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-453: ARMCortexR52PlusCT MTI instances

InstanceName	ComponentName
ARMCortexR52PlusCT	ARMv8Cluster
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.l1licache	PVCache

InstanceName	ComponentName
ARMCortexR52PlusCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR52PlusCT.ext_bus	PVBusLogger
ARMCortexR52PlusCT.ext_bus.mapper	PVBusMapper
ARMCortexR52PlusCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR52PlusCT.gic_iri	GICv3IRI
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

ARMCortexR52PlusCT contains the following CADI targets:

- ARM_Cortex-R52Plus
- Cluster_ARM_Cortex-R52Plus
- PVCache
- TlbCadi
- gic_iri

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.

Ports for ARMCortexR52PlusCT

Table 3-454: Ports

Name	Protocol	Type	Description
cfgdbgromaddr	Value_64	Slave	Debug ROM base address.

Name	Protocol	Type	Description
cfgdbgromaddrv	Signal	Slave	Debug ROM base address valid.
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.
commr[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 COREACTIVEx
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.

Name	Protocol	Type	Description
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.
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 ARMCortexR52PlusCT

CLUSTER_ID

Type

int

Default value

0x0

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

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

Equivalent to CFGDBGROMADDR.

DBGROMADDRV**Type**

bool

Default value

0x0

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x0

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

Number of cores in cluster.

PERIPHBASE**Type**

int

Default value

0x13080000

Equivalent to CFGPERIPHBASE.

cluster_utid**Type**

int

Default value

0x0

Equivalent to CFGCLUSTERUTID.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Equivalent to CFGTHUMBEXCEPTIONS.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Equivalent to CFGVECTABLE.

cpuX.ase-present**Type**

bool

Default value

0x1

Set whether the model has been built with NEON support.

cpuX.dcache-size**Type**

int

Default value

0x8000

L1 D-Cache size in bytes.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.flash.enable**Type**

bool

Default value

0x0

Equivalent to CFGFLASHEN.

cpuX.icache-size**Type**

int

Default value

0x8000

L1 I-Cache size in bytes.

cpuX.llpp.base**Type**

int

Default value

0x0

Equivalent to CFGLLPPBASEADDR.

cpuX.llpp.size**Type**

int

Default value

0x8000000

Equivalent to CFGLLPPSIZE.

cpuX.max_code_cache_mb**Type**

int

Default value

0x100

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.tcm.a.enable**Type**

bool

Default value

0x0

Equivalent to CFGTCMBOOT.

cpuX.tcm.a.size**Type**

int

Default value

0x4000

Sets the size of the ATCM(in bytes).

cpuX.tcm.a.wait**Type**

int

Default value

0x0

TCM Register A accesses wait states: 0-1 states.

cpuX.tcm.b.size**Type**

int

Default value

0x4000

Sets the size of the BTCM(in bytes).

cpuX.tcm.b.wait**Type**

int

Default value

0x0

TCM Register B accesses wait states: 0-1 states.

cpuX.tcm.c.size**Type**

int

Default value

0x2000

Sets the size of the CTCM(in bytes).

cpuX.tcm.c.wait**Type**

int

Default value

0x0

TCM Register C accesses wait states: 0-1 states.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

Whether double-precision floating point feature is implemented (FEAT_F64MM).

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

(equivalent to `CFGSLSPPLIT`).

`enable_simulation_performance_optimizations`

Type

bool

Default value

0x1

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

Equivalent to `CFGFLASHPROTEN`.

`has_export_m_port`

Type

bool

Default value

0x1

The interrupt distributor has an optional interrupt export port for routing interrupts to an external device.

has_flash**Type**

bool

Default value

0x0

Equivalent to CFGFLASHIMP.

has_flash_protection**Type**

bool

Default value

0x1

Equivalent to CFGFLASHPROTIMP.

has_llpp**Type**

bool

Default value

0x0

Equivalent to CFGLLPPIMP.

icache-prefetch_enabled**Type**

bool

Default value

0x0

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

Set whether I-cache has stateful implementation.

`memory.ext_slave_base`**Type**

int

Default value

0x0

Equivalent to `CFGAXISTCMBASEADDR`.

`memory.flash_base`**Type**

int

Default value

0x0

Equivalent to `CFGFLASHBASEADDR`.

`num_protection_regions_s1`**Type**

int

Default value

0x18

Number of v8-R protection regions.

`num_protection_regions_s2`**Type**

int

Default value

0x18

Number of v8-R hyp protection regions.

num_spi**Type**

int

Default value

0x3c0

Number of interrupts (SPI) into the internal GIC controller.

ram_protection_enable_at_reset**Type**

bool

Default value

0x0

Equivalent to CFGRAMPROTEN.

reported_fp_revision**Type**

int

Default value

0xffffffffffffffff

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

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

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.

3.5.66 ARM Cortex R52x1CT

ARM Cortex R52x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-455: IP revisions support

Revision	Quality level
r1p4	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Changes in 11.27.17

Parameters added:

- `has_llpp`

Iris and MTI instances for ARM Cortex R52x1CT

This model has the following Iris instances:

Table 3-456: ARM Cortex R52x1CT Iris instances

InstanceName	ComponentName
ARM Cortex R52x1CT	Cluster_ARM_Cortex-R52
ARM Cortex R52x1CT.AMU	PVBusLogger
ARM Cortex R52x1CT.AMU.mapper	PVBusMapper
ARM Cortex R52x1CT.DAP	PVBusLogger
ARM Cortex R52x1CT.DAP.mapper	PVBusMapper
ARM Cortex R52x1CT.MMAP	PVBusLogger
ARM Cortex R52x1CT.MMAP.mapper	PVBusMapper
ARM Cortex R52x1CT.RAS	PVBusLogger
ARM Cortex R52x1CT.RAS.mapper	PVBusMapper
ARM Cortex R52x1CT.acp_mapper	PVBusMapper
ARM Cortex R52x1CT.cpu0	ARM_CortexR52
ARM Cortex R52x1CT.cpu0.UTLB	TLB
ARM Cortex R52x1CT.cpu0.dtlb	TlbCadi
ARM Cortex R52x1CT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARM Cortex R52x1CT.cpu0.l1dcache	PVCache
ARM Cortex R52x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM Cortex R52x1CT.cpu0.l1licache	PVCache
ARM Cortex R52x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARM Cortex R52x1CT.ext_bus	PVBusLogger
ARM Cortex R52x1CT.ext_bus.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexR52x1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR52x1CT.gic_iri	GIC_IRI
ARMCortexR52x1CT.gic_iri.rd_0	GICv3RedistributorInternal
ARMCortexR52x1CT.gic_iri.rd_0_0	GICv3RedistributorInternal
ARMCortexR52x1CT.gic_iri.rd_0_0_0	GICv3RedistributorInternal
ARMCortexR52x1CT.gic_iri.rd_0_0_0_0	GICv3Redistributor
ARMCortexR52x1CT.gic_iri.rd_0_0_0_1	GICv3Redistributor
ARMCortexR52x1CT.gic_iri.rd_tl	GICv3Distributor
ARMCortexR52x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Table 3-457: ARMCortexR52x1CT MTI instances

InstanceName	ComponentName
ARMCortexR52x1CT	ARMv8Cluster
ARMCortexR52x1CT.AMU	PVBusLogger
ARMCortexR52x1CT.AMU.mapper	PVBusMapper
ARMCortexR52x1CT.DAP	PVBusLogger
ARMCortexR52x1CT.DAP.mapper	PVBusMapper
ARMCortexR52x1CT.MMAP	PVBusLogger
ARMCortexR52x1CT.MMAP.mapper	PVBusMapper
ARMCortexR52x1CT.RAS	PVBusLogger
ARMCortexR52x1CT.RAS.mapper	PVBusMapper
ARMCortexR52x1CT.acp_mapper	PVBusMapper
ARMCortexR52x1CT.cpu0	ARM_CortexR52
ARMCortexR52x1CT.cpu0.UTLB	TLB
ARMCortexR52x1CT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR52x1CT.cpu0.l1dcache	PVCache
ARMCortexR52x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR52x1CT.cpu0.l1licache	PVCache
ARMCortexR52x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexR52x1CT.ext_bus	PVBusLogger
ARMCortexR52x1CT.ext_bus.mapper	PVBusMapper
ARMCortexR52x1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR52x1CT.gic_iri	GICv3IRI
ARMCortexR52x1CT.gic_iri.rd_0	GICv3RedistributorInternal
ARMCortexR52x1CT.gic_iri.rd_0_0	GICv3RedistributorInternal
ARMCortexR52x1CT.gic_iri.rd_0_0_0	GICv3RedistributorInternal
ARMCortexR52x1CT.gic_iri.rd_0_0_0_0	GICv3Redistributor
ARMCortexR52x1CT.gic_iri.rd_0_0_0_1	GICv3Redistributor

InstanceName	ComponentName
ARMCortexR52x1CT.gic_iri.rd_tl	GICv3Distributor
ARMCortexR52x1CT.l2_flusher	AsyncCacheFlushUnit

ARMCortexR52x1CT contains the following CADI targets:

- ARM_CortexR52
- Cluster_ARM_Cortex-R52
- PVCache
- TlbCadi
- gic_iri

About ARMCortexR52x1CT

- The following components also exist:
 - ARMCortexR52x2CT.
 - ARMCortexR52x3CT.
 - ARMCortexR52x4CT.

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.

Debug accesses

For debug accesses to succeed, they must have permissions that are compliant with the permission value in the `IMP_SLAVEPCTLR` register.



The reset value of IMP_SLAVEPCTLR is 0x1 which means privileged access only.

Ports for ARMCortexR52x1CT

Table 3-458: Ports

Name	Protocol	Type	Description
cfgdbgromaddr	Value_64	Slave	Debug ROM base address.
cfgdbgromaddrv	Signal	Slave	Debug ROM base address valid.
cfgendianess[1]	Signal	Slave	This signal is for EE bit initialisation.
cfgperiphbase	Value_64	Slave	This port sets the base address of private peripheral region.
cfgthumbexceptions[1]	Signal	Slave	This signal provides default exception handling state.
cfgvectable[1]	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]
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
commr[1]	Signal	Master	Receive portion of Data Transfer Register full.
commtx[1]	Signal	Master	Transmit portion of Data Transfer Register empty.
COREPACTIVEx1[1]	Signal	Master	These signals relate to core power down. Equivalent to COREPACTIVEx[1]
cpuhalt[1]	Signal	Slave	Raising this signal will put the core into halt mode.
cpuporeset[1]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cti[1]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
dbgack[1]	Signal	Master	External debug interface.
dbgen[1]	Signal	Slave	External debug interface.
dbgnopwrdown[1]	Signal	Master	There is no support for PChannel in CortexR52. These signals relate to core power down. Equivalent to COREPACTIVEx[0]
dbgpwrupreq[1]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.

Name	Protocol	Type	Description
edbgrq[1]	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.
flash_m[1]	PVBus	Master	Flash Port.
gdu_external_m	GICv3Comms	Master	GDU external messaging port.
hiden[1]	Signal	Slave	External debug interface.
hniden[1]	Signal	Slave	External debug interface.
llpp_m[1]	PVBus	Master	LLPP (Low-Latency Peripheral Port).
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[1]	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[1]	PVBus	Master	The core will generate bus requests on this port. Equivalent to AXIM port
reset[1]	Signal	Slave	Raising this signal will put the core into reset mode.
sei[1]	Signal	Slave	Per core virtual System Error physical pins.
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
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.
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.
topreset	Signal	Slave	This signal resets timer and interrupt controller.
vsei[1]	Signal	Slave	Per core virtual System Error physical pins.
warmrstreq[1]	Signal	Master	Warm reset request from core.

Parameters for ARMCortexR52x1CT

CLUSTER_ID

Type

int

Default value

0x0

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

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

Equivalent to CFGDBGROMADDR.

DBGROMADDRV**Type**

bool

Default value

0x0

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x0

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

PERIPHBASE**Type**

int

Default value

0x13080000

Equivalent to CFGPERIPHBASE.

cluster_utid**Type**

int

Default value

0x0

Equivalent to CFGCLUSTERUTID.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Equivalent to CFGTHUMBEXCEPTIONS.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Equivalent to CFGVECTABLE.

cpuX.ase-present**Type**

bool

Default value

0x1

Set whether the model has been built with NEON support.

cpuX.dcache-size**Type**

int

Default value

0x8000

L1 D-Cache size in bytes.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.flash.enable**Type**

bool

Default value

0x0

Equivalent to CFGFLASHEN.

cpuX.icache-size**Type**

int

Default value

0x8000

L1 I-Cache size in bytes.

cpuX.llpp.base**Type**

int

Default value

0x0

Equivalent to CFGLLPPBASEADDR.

cpuX.llpp.size**Type**

int

Default value

0x1000

Equivalent to CFGLLPPSIZE.

cpuX.max_code_cache_mb**Type**

int

Default value

0x100

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.tcm.a.enable**Type**

bool

Default value

0x0

Equivalent to CFGTCMBOOT.

cpuX.tcm.a.size**Type**

int

Default value

0x4000

Sets the size of the ATCM(in bytes).

cpuX.tcm.a.wait**Type**

int

Default value

0x0

TCM Register A accesses wait states: 0-1 states.

cpuX.tcm.b.size**Type**

int

Default value

0x4000

Sets the size of the BTCM(in bytes).

cpuX.tcm.b.wait**Type**

int

Default value

0x0

TCM Register B accesses wait states: 0-1 states.

cpuX.tcm.c.size**Type**

int

Default value

0x2000

Sets the size of the CTCM(in bytes).

cpuX.tcm.c.wait**Type**

int

Default value

0x0

TCM Register C accesses wait states: 0-1 states.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

Whether double-precision floating point feature is implemented (FEAT_F64MM).

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

(equivalent to `CFGSLSPPLIT`).

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Equivalent to `CFGFLASHPROTEN`.

has_export_m_port**Type**

bool

Default value

0x1

The interrupt distributor has an optional interrupt export port for routing interrupts to an external device.

has_flash**Type**

bool

Default value

0x0

Equivalent to CFGFLASHIMP.

has_flash_protection**Type**

bool

Default value

0x1

Equivalent to CFGFLASHPROTIMP.

has_llpp**Type**

bool

Default value

0x0

Equivalent to CFGLLPPIMP.

icache-prefetch_enabled**Type**

bool

Default value

0x0

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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`

Set whether I-cache has stateful implementation.

`memory.ext_slave_base`**Type**`int`**Default value**`0x0`

Equivalent to `CFGAXISTCMBASEADDR`.

`memory.flash_base`**Type**`int`**Default value**`0x0`

Equivalent to `CFGFLASHBASEADDR`.

`num_protection_regions_s1`**Type**`int`**Default value**`0x18`

Number of v8-R protection regions.

`num_protection_regions_s2`**Type**`int`**Default value**`0x18`

Number of v8-R hyp protection regions.

num_spi**Type**

int

Default value

0x3c0

Number of interrupts (SPI) into the internal GIC controller.

ram_protection_enable_at_reset**Type**

bool

Default value

0x0

Equivalent to CFGRAMPROTEN.

reported_fp_revision**Type**

int

Default value

0xffffffffffffffff

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

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

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 ARM Cortex R82AECT

ARM Cortex R82AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-459: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM Cortex R82AECT

This model has the following Iris instances:

Table 3-460: ARM Cortex R82AECT Iris instances

InstanceName	ComponentName
ARM Cortex R82AECT	Cluster_ARM_Cortex-R82AE
ARM Cortex R82AECT.AMU	PVBusLogger
ARM Cortex R82AECT.AMU.mapper	PVBusMapper
ARM Cortex R82AECT.DAP	PVBusLogger
ARM Cortex R82AECT.DAP.mapper	PVBusMapper
ARM Cortex R82AECT.DSU	DSU
ARM Cortex R82AECT.DSU.PPU_cluster	PPUv1
ARM Cortex R82AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARM Cortex R82AECT.DSU.PPU_core0	PPUv1
ARM Cortex R82AECT.DSU.PPU_core0.busslave	PVBusSlave
ARM Cortex R82AECT.DSU.utility_slave[0]	PVBusSlave
ARM Cortex R82AECT.MMAP	PVBusLogger
ARM Cortex R82AECT.MMAP.mapper	PVBusMapper
ARM Cortex R82AECT.RAS	PVBusLogger
ARM Cortex R82AECT.RAS.mapper	PVBusMapper
ARM Cortex R82AECT.cpu0	ARM_Cortex-R82AE
ARM Cortex R82AECT.cpu0.UTLB	TLB
ARM Cortex R82AECT.cpu0.dtlb	TlbCadi
ARM Cortex R82AECT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARM Cortex R82AECT.cpu0.l1dcache	PVCache
ARM Cortex R82AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM Cortex R82AECT.cpu0.l1icache	PVCache
ARM Cortex R82AECT.cpu0.l1icache.upstream[0]	PVBusSlave
ARM Cortex R82AECT.ext_bus	PVBusLogger

InstanceName	ComponentName
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
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

This model has the following MTI trace components:

Table 3-461: ARMCortexR82AECT MTI instances

InstanceName	ComponentName
ARMCortexR82AECT	ARMv8Cluster
ARMCortexR82AECT.AMU	PVBusLogger
ARMCortexR82AECT.AMU.mapper	PVBusMapper
ARMCortexR82AECT.DAP	PVBusLogger
ARMCortexR82AECT.DAP.mapper	PVBusMapper

InstanceName	ComponentName
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
ARMCortexR82AECT.l2_cache.upstream[8]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[9]	PVBusSlave
ARMCortexR82AECT.llram_atomic_filter	LLRAMAtomicOpFilter

InstanceName	ComponentName
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 contains the following CADI targets:

- ARM_Cortex-R82AE
- Cluster_ARM_Cortex-R82AE
- PVCache
- TlbCadi

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.

Ports for ARMCortexR82AECT

Table 3-462: 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

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 core executes instructions.
cluster_pcs_m_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_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.

Name	Protocol	Type	Description
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 I2cache
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).

Name	Protocol	Type	Description
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 ARM Cortex R82AECT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation to NC/DEV memory. The broadcastatomic signal will override this value if used.

BROADCASTATOMICL

Type

bool

Default value

0x1

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

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CCSIDR-L1D_override**Type**

int

Default value

0x0

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

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

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

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

CFGTFPEN pin at reset.

CHI**Type**

bool

Default value

0x0

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

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

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

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x0

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x0

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

Number of cores in cluster.

PA_SIZE**Type**

int

Default value

0x28

Physical address range supported (FEAT_LPA).

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

VMSA_supported**Type**

bool

Default value

0x1

VMSA is supported at EL1.

bus_protection_enable_at_reset**Type**

bool

Default value

0x0

Equivalent to CFGBUSPROTEN.

core_power_on_by_default**Type**

bool

Default value

0x1

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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.RVBAR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.TEINIT**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.ase-present**Type**

bool

Default value

0x1

Set whether the model has been built with NEON support.

cpuX.dcache-size**Type**

int

Default value

0x10000

L1 D-Cache size in bytes.

cpuX.dtcn_base**Type**

int

Default value

0x0

Sets the 16K aligned base address of DTCM.

cpuX.dtcn_size**Type**

int

Default value

0x0

Sets the size of DTCM (in bytes).

cpuX.dtcn_stretch_clk**Type**

bool

Default value

0x0

Whether DTCM clock stretched to occupy full cycle.

cpuX.dtcm_wait**Type**

int

Default value

0x0

DTCM accesses wait states: 0-3 cycles.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.icache-size**Type**

int

Default value

0x10000

L1 I-Cache size in bytes.

cpuX.itcm_base**Type**

int

Default value

0x0

Sets the 16K aligned base address of ITCM.

cpuX.itcm_size**Type**

int

Default value

0x0

Sets the size of ITCM (in bytes).

cpuX.itcm_stretch_clk**Type**

bool

Default value

0x0

Whether ITCM clock stretched to occupy full cycle.

cpuX.itcm_wait**Type**

int

Default value

0x0

ITCM accesses wait states: 0-3 cycles.

cpuX.llpp.base**Type**

int

Default value

0x0

Equivalent to CFGLLPPBASEADDR.

cpuX.llpp.size**Type**

int

Default value

0x8000000

Equivalent to CFGLLPPSIZE.

cpuX.max_code_cache_mb**Type**

int

Default value

0x100

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.tcm.a.enable**Type**

bool

Default value

0x0

Equivalent to CFGITCMENm.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

Whether double-precision floating point feature is implemented (FEAT_F64MM).

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`**Type**

int

Default value

0x0

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

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

Whether the core is configured in Dual Core Lock Step mode (FEAT_DCLS).

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

The minimum value for the GICC_BPR register (non-secure version will be 1 + this value).

gicv3.VBPR-min**Type**

int

Default value

0x2

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

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

Support the Transient Fault Protection (TFP) flop parity errors through RAS registers (FEAT_TFP).

has_llpp**Type**

bool

Default value

0x1

Equivalent to CFGLLPPIMP.

has_pmc**Type**

bool

Default value

0x0

Programmable MBIST controllers implemented.

has_spp**Type**

bool

Default value

0x1

Equivalent to CFGSPPIMP.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

Set whether I-cache has stateful implementation.

l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`l2cache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

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

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

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

Equivalent to CFGACELSTCMBASEADDR.

memory.has_llram**Type**

bool

Default value

0x1

Equivalent to CFGLLRAMIMP.

memory.llram_base**Type**

int

Default value

0x20000000

Equivalent to CFGLLRAMBASEADDR.

memory.llram_enable_at_reset**Type**

bool

Default value

0x1

Equivalent to CFGLLRAMEN.

memory.llram_shared**Type**

bool

Default value

0x0

Equivalent to CFGLLRAMSHARED and it is only functional for revision 2.

memory.llram_size**Type**

int

Default value

0x10000000

Size of the LLRAM.

num_protection_regions_s1**Type**

int

Default value

0x10

Number of v8-R protection regions.

num_protection_regions_s2**Type**

int

Default value

0x10

Number of v8-R hyp protection regions.

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

ram_protection_enable_at_reset**Type**

bool

Default value

0x0

Equivalent to CFGGRAMPROTEN.

spp.base**Type**

int

Default value

0x0

Equivalent to CFGSPPBASEADDR.

spp.size**Type**

int

Default value

0x8000000

Sets the size of SPP(in bytes).

stage12_tlb_size**Type**

int

Default value

0x0

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

walk_cache_latency**Type**

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.68 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 3-463: 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 23.

Iris and MTI instances for ARM CortexR82CT

This model has the following Iris instances:

Table 3-464: ARM CortexR82CT Iris instances

InstanceName	ComponentName
ARM CortexR82CT	Cluster_ARM_Cortex-R82
ARM CortexR82CT.AMU	PVBusLogger
ARM CortexR82CT.AMU.mapper	PVBusMapper
ARM CortexR82CT.DAP	PVBusLogger
ARM CortexR82CT.DAP.mapper	PVBusMapper
ARM CortexR82CT.DSU	DSU

InstanceName	ComponentName
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.dtlb	TlbCadi
ARMCortexR82CT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR82CT.cpu0.l1dcache	PVCache
ARMCortexR82CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR82CT.cpu0.l1licache	PVCache
ARMCortexR82CT.cpu0.l1licache.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
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.l1ram_atomic_filter	LLRAMAtomicOpFilter

InstanceName	ComponentName
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

This model has the following MTI trace components:

Table 3-465: ARMCortexR82CT MTI instances

InstanceName	ComponentName
ARMCortexR82CT	ARMv8Cluster
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.l1licache	PVCache
ARMCortexR82CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexR82CT.ext_bus	PVBusLogger
ARMCortexR82CT.ext_bus.mapper	PVBusMapper
ARMCortexR82CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR82CT.l2_cache	PVCache

InstanceName	ComponentName
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 contains the following CADI targets:

- ARM_Cortex-R82
- Cluster_ARM_Cortex-R82
- PVCache
- TlbCadi

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.

Ports for ARMCortexR82CT

Table 3-466: 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_pcs_m_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
CNTHVSIQ[8]	Signal	Master	Timer signals to SOC
CNTPSIQ[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_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.

Name	Protocol	Type	Description
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 I2cache
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

Name	Protocol	Type	Description
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 ARM Cortex R82CT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation to NC/DEV memory. The broadcastatomic signal will override this value if used.

BROADCASTATOMICL**Type**

bool

Default value

0x1

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

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

CCSIDR-L1D_override**Type**

int

Default value

0x0

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

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

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

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

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

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

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

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

DBGROMADDRV**Type**

bool

Default value

0x0

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

GICDISABLE**Type**

bool

Default value

0x0

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

Number of cores in cluster.

PA_SIZE**Type**

int

Default value

0x28

Physical address range supported (FEAT_LPA).

PERIPHBASE**Type**

int

Default value

0x13080000

Base address of peripheral memory space.

VMSA_supported**Type**

bool

Default value

0x0

VMSA is supported at EL1.

core_power_on_by_default**Type**

bool

Default value

0x1

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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CP15SDISABLE**Type**

bool

Default value

0x0

Initialize to disable access to some CP15 registers.

cpuX.RVBAR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.TEINIT**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.ase-present**Type**

bool

Default value

0x1

Set whether the model has been built with NEON support.

cpuX.dcache-size**Type**

int

Default value

0x10000

L1 D-Cache size in bytes.

cpuX.dtcm_base**Type**

int

Default value

0x0

Sets the 16K aligned base address of DTCM.

cpuX.dtcm_size**Type**

int

Default value

0x0

Sets the size of DTCM (in bytes).

cpuX.dtcm_stretch_clk**Type**

bool

Default value

0x0

Whether DTCM clock stretched to occupy full cycle.

cpuX.dtcn_wait**Type**

int

Default value

0x0

DTCM accesses wait states: 0-3 cycles.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.icache-size**Type**

int

Default value

0x10000

L1 I-Cache size in bytes.

cpuX.itcm_base**Type**

int

Default value

0x0

Sets the 16K aligned base address of ITCM.

cpuX.itcm_size**Type**

int

Default value

0x0

Sets the size of ITCM (in bytes).

cpuX.itcm_stretch_clk**Type**

bool

Default value

0x0

Whether ITCM clock stretched to occupy full cycle.

cpuX.itcm_wait**Type**

int

Default value

0x0

ITCM accesses wait states: 0-3 cycles.

cpuX.llpp.base**Type**

int

Default value

0x0

Equivalent to CFGLLPPBASEADDR.

cpuX.llpp.size**Type**

int

Default value

0x8000000

Equivalent to CFGLLPPSIZE.

cpuX.max_code_cache_mb**Type**

int

Default value

0x100

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.tcm.a.enable**Type**

bool

Default value

0x0

Equivalent to CFGITCMENm.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

Whether double-precision floating point feature is implemented (FEAT_F64MM).

cpuX.vfp-enable_at_reset**Type**

bool

Default value

0x0

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`**Type**

int

Default value

0x0

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

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

Whether the core is configured in Dual Core Lock Step mode (FEAT_DCLS).

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

The minimum value for the GICC_BPR register (non-secure version will be 1 + this value).

gicv3.VBPR-min**Type**

int

Default value

0x2

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

If true, the cluster follows the dense memory map else it implements the sparse memory map.

has_llpp**Type**

bool

Default value

0x1

Equivalent to CFGLLPPIMP.

has_spp**Type**

bool

Default value

0x1

Equivalent to CFGSPPIMP.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

Set whether I-cache has stateful implementation.

l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Equivalent to CFGACELSTCMBASEADDR.

memory.has_llram**Type**

bool

Default value

0x1

Equivalent to CFGLLRAMIMP.

memory.llram_base**Type**

int

Default value

0x20000000

Equivalent to CFGLLRAMBASEADDR.

memory.llram_enable_at_reset**Type**

bool

Default value

0x1

Equivalent to CFGLLRAMEN.

memory.llram_shared**Type**

bool

Default value

0x0

Equivalent to CFGLLRAMSHARED and it is only functional for revision 2.

memory.llram_size**Type**

int

Default value

0x10000000

Size of the LLRAM.

num_protection_regions_s1**Type**

int

Default value

0x10

Number of v8-R protection regions.

num_protection_regions_s2**Type**

int

Default value

0x10

Number of v8-R hyp protection regions.

patch_level**Type**

int

Default value

0x1

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

ram_protection_enable_at_reset**Type**

bool

Default value

0x0

Equivalent to CFGRAMPROTEN.

revision_number**Type**

int

Default value

0x1

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

Equivalent to CFGSPPBASEADDR.

spp.size**Type**

int

Default value

0x8000000

Sets the size of SPP(in bytes).

stage12_tlb_size**Type**

int

Default value

0x0

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

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

walk_cache_latency**Type**

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.69 ARM CortexX1CCT

ARM CortexX1CCT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-467: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM CortexX1CCT

This model has the following Iris instances:

Table 3-468: ARM CortexX1CCT Iris instances

InstanceName	ComponentName
ARM CortexX1CCT	Cluster_ARM_Cortex-X1C
ARM CortexX1CCT.AMU	PVBusLogger
ARM CortexX1CCT.AMU.mapper	PVBusMapper
ARM CortexX1CCT.DAP	PVBusLogger
ARM CortexX1CCT.DAP.mapper	PVBusMapper
ARM CortexX1CCT.DSU	DSU
ARM CortexX1CCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM CortexX1CCT.DSU.mpam_busslave	PVBusSlave
ARM CortexX1CCT.DSU.shared_cache	PVCache
ARM CortexX1CCT.DSU.shared_cache.upstream[0]	PVBusSlave
ARM CortexX1CCT.DSU.shared_cache.upstream[1]	PVBusSlave
ARM CortexX1CCT.DSU.shared_cache.upstream[2]	PVBusSlave
ARM CortexX1CCT.DSU.shared_cache.upstream[3]	PVBusSlave
ARM CortexX1CCT.MMAP	PVBusLogger
ARM CortexX1CCT.MMAP.mapper	PVBusMapper
ARM CortexX1CCT.RAS	PVBusLogger
ARM CortexX1CCT.RAS.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexX1CCT.cpu0	ARM_Cortex-X1C
ARMCortexX1CCT.cpu0.UTLB	TLB
ARMCortexX1CCT.cpu0.dtlb	TlbCadi
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

This model has the following MTI trace components:

Table 3-469: ARMCortexX1CCT MTI instances

InstanceName	ComponentName
ARMCortexX1CCT	ARMv8Cluster
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

InstanceName	ComponentName
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 contains the following CADI targets:

- ARM_Cortex-X1C
- Cluster_ARM_Cortex-X1C
- PVCache
- TlbCadi

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.



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](#).

Ports for ARMCortexX1CCT

Table 3-470: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>AENDMP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>ASTARTMP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>broadcastatomic</code>	Signal	Slave	CHI defined pins.
<code>broadcastcachemaint</code>	Signal	Slave	ACE defined pins.
<code>broadcastouter</code>	Signal	Slave	ACE defined pins.
<code>broadcastpersist</code>	Signal	Slave	CHI defined pins.
<code>cfgend[8]</code>	Signal	Slave	This signal is for EE bit initialisation.
<code>cfgsdisable</code>	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
<code>cfgte[8]</code>	Signal	Slave	This signal provides default exception handling state.
<code>clk_in</code>	ClockSignal	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
<code>clusterid</code>	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
<code>clusterpmuirq</code>	Signal	Master	DynamlQ pmu irq
<code>CNTHPIRQ[8]</code>	Signal	Master	Timer signals to SOC.
<code>CNTHVIRQ[8]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPNSIRQ[8]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPSIRQ[8]</code>	Signal	Master	Timer signals to SOC.
<code>cntvalueb</code>	CounterInterface	Slave	Interface to SoC level counter module.
<code>CNTVIRQ[8]</code>	Signal	Master	Timer signals to SOC.

Name	Protocol	Type	Description
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.
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

Name	Protocol	Type	Description
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 ARMCortexX1CCT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

If true, Accelerator Coherency Port is configured.

has_coherent_icache**Type**

bool

Default value

0x0

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

`icache-state_modelled`

Type

bool

Default value

0x0

Set whether I-cache has stateful implementation.

`l3cache-hit_latency`

Type

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end

Type

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start

Type

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency

Type

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency

Type

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.70 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 3-471: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexX1CT

This model has the following Iris instances:

Table 3-472: 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.dtlb	TlbCadi
ARMCortexX1CT.cpu0.l1dcache	PVCache
ARMCortexX1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX1CT.cpu0.l1licache	PVCache
ARMCortexX1CT.cpu0.l1licache.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

This model has the following MTI trace components:

Table 3-473: ARMCortexX1CT MTI instances

InstanceName	ComponentName
ARMCortexX1CT	ARMv8Cluster
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.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 contains the following CADI targets:

- ARM_Cortex-X1
- Cluster_ARM_Cortex-X1
- PVCache
- TlbCadi

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 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](#).

Ports for ARMCortexX1CT

Table 3-474: 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 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.
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.

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[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.

Name	Protocol	Type	Description
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 CortexX1CT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

`dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`

Type

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

`enable_simulation_performance_optimizations`**Type**

bool

Default value

0x1

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

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

If true, Accelerator Coherency Port is configured.

has_coherent_icache**Type**

bool

Default value

0x0

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

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

`icache-state_modelled`**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

`l3cache-hit_latency`**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end**Type**

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.71 ARM CortexX2CT

ARM CortexX2CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-475: IP revisions support

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM CortexX2CT

This model has the following Iris instances:

Table 3-476: ARM CortexX2CT Iris instances

InstanceName	ComponentName
ARM CortexX2CT	Cluster_ARM_Cortex-X2
ARM CortexX2CT.AMU	PVBusLogger
ARM CortexX2CT.AMU.mapper	PVBusMapper
ARM CortexX2CT.DAP	PVBusLogger
ARM CortexX2CT.DAP.mapper	PVBusMapper
ARM CortexX2CT.DSU	DSU
ARM CortexX2CT.DSU.PPU_cluster	PPUv1

InstanceName	ComponentName
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.dtlb	TlbCadi
ARMCortexX2CT.cpu0.l1dcache	PVCache
ARMCortexX2CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX2CT.cpu0.l1icache	PVCache
ARMCortexX2CT.cpu0.l1icache.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	GLCv3CPUInterfaceDecoder
ARMCortexX2CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-477: ARMCortexX2CT MTI instances

InstanceName	ComponentName
ARMCortexX2CT	ARMv8Cluster
ARMCortexX2CT.AMU	PVBusLogger
ARMCortexX2CT.AMU.mapper	PVBusMapper
ARMCortexX2CT.DAP	PVBusLogger
ARMCortexX2CT.DAP.mapper	PVBusMapper
ARMCortexX2CT.DSU	DSU
ARMCortexX2CT.DSU.PPU_cluster	PPUv1

InstanceName	ComponentName
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.l1icache	PVCache
ARMCortexX2CT.cpu0.l1icache.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

ARMCortexX2CT contains the following CADI targets:

- ARM_Cortex-X2
- Cluster_ARM_Cortex-X2
- PVCache
- TlbCadi

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](#).

Ports for ARMCortexX2CT

Table 3-478: 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.

Name	Protocol	Type	Description
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_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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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 CortexX2CT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

core_power_on_by_default**Type**

bool

Default value

0x0

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 PPUs 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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`

Type

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Number of claim tags.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.REVISION**Type**

int

Default value

0x2

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

ext_abort_device_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

If true, Accelerator Coherency Port is configured.

has_ete**Type**

bool

Default value

0x0

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

If true, an additional AXI peripheral port is configured.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`

Type

int

Default value

0x0

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

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

Set whether I-cache has stateful implementation.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency

Type

int

Default value

0x0

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

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

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

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

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

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

Number of transport nodes. Zero implies direct-connect configuration.

pchannel_treat_simreset_as_poreset**Type**

bool

Default value

0x0

Register core as ON state to cluster with simulation reset.

periph_address_end**Type**

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.72 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 3-479: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexX3CT

This model has the following Iris instances:

Table 3-480: 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.dtlb	TlbCadi
ARMCortexX3CT.cpu0.l1dcache	PVCache
ARMCortexX3CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX3CT.cpu0.l1icache	PVCache
ARMCortexX3CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX3CT.cpu0.l2cache	PVCache

InstanceName	ComponentName
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.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-481: ARMCortexX3CT MTI instances

InstanceName	ComponentName
ARMCortexX3CT	ARMv8Cluster
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
ARMCortexX3CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX3CT.cpu0.l1licache	PVCache
ARMCortexX3CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexX3CT.cpu0.l2cache	PVCache

InstanceName	ComponentName
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 contains the following CADI targets:

- ARM_Cortex-X3
- Cluster_ARM_Cortex-X3
- PVCache
- TlbCadi

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.



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.

Ports for ARMCortexX3CT

Table 3-482: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>AEND0MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND1MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND2MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND3MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>ASTART0MP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>ASTART1MP</code>	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[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_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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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_pcsmp_channel[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.
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.

Name	Protocol	Type	Description
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 ARMCortexX3CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Value of CPU Configuration Register.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

bus_type**Type**

int

Default value

0x0

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

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 PPUs 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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency

Type

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

`dcache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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}]]"
```

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

Number of claim tags.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CMOD value.

ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCRSRTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCRSR.TA value for a forcibly traced exception.

ext_abort_normal_noncacheable_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of normal noncacheable-read external aborts.

ext_abort_so_write_ras_type**Type**

int

Default value

0x2

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

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

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

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 MPAMO_EL1, MPAM1_EL1, MPAM2_EL2, MPAM3_EL3, MPAMHCR_EL2.

has_enhanced_pan**Type**

int

Default value

0x2

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

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

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

If true, an additional AXI peripheral port is configured.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

l3cache-has_mpam**Type**

bool

Default value

0x1

L3 Cache has MPAM support.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L3 Cache number of ways (sets are implicit from size).

`l3cache-write_access_latency`

Type

int

Default value

0x0

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

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

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

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

Number of transport nodes. Zero implies direct-connect configuration.

pmu-num_counters**Type**

int

Default value

0x6

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}]"
```

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

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.73 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 3-483: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMCortexX4CT

This model has the following Iris instances:

Table 3-484: ARMCortexX4CT Iris instances

InstanceName	ComponentName
ARMCortexX4CT	Cluster_ARM_Cortex-X4
ARMCortexX4CT.AMU	PVBusLogger
ARMCortexX4CT.AMU.mapper	PVBusMapper

InstanceName	ComponentName
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.dtlb	TlbCadi
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
ARMCortexX4CT.ext_bus.mapper	PVBusMapper
ARMCortexX4CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexX4CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-485: ARMCortexX4CT MTI instances

InstanceName	ComponentName
ARMCortexX4CT	ARMv8Cluster

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
ARMCortexX4CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX4CT.ext_bus	PVBusLogger
ARMCortexX4CT.ext_bus.mapper	PVBusMapper
ARMCortexX4CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

ARMCortexX4CT contains the following CADI targets:

- ARM_Cortex-X4
- Cluster_ARM_Cortex-X4
- PVCache

- TlbCadi

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 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](#).

Limitations

- Some of the SVE instructions do not raise undefined exceptions when SP is used as a source register.

Ports for ARMCortexX4CT

Table 3-486: 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_pcsmpchannel	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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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 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.
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.

Name	Protocol	Type	Description
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.
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).

Name	Protocol	Type	Description
vsei[14]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM CortexX4CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

Value of CPU Configuration Register.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

bus_type**Type**

int

Default value

0x0

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

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 PPUs 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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX . CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX . CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX . CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

ecv_support_level**Type**

int

Default value

0x2

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

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

Number of claim tags.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

ext_abort_so_write_ras_type**Type**

int

Default value

0x2

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

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

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

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 MPAMO_EL1, MPAM1_EL1, MPAM2_EL2, MPAM3_EL3, MPAMHCR_EL2.

has_enhanced_pan**Type**

int

Default value

0x2

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

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

If true, an additional AXI peripheral port is configured.

has_statistical_profiling**Type**

bool

Default value

0x1

Whether Statistical Based Profiling is implemented (FEAT_SPE).

has_v8_7_spe_inverted_filtering**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the inverting filtering by events.

has_v8_7_spe_previous_branch_target**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the optional branch target feature is implemented.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`

Type

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

l3cache-size**Type**

int

Default value

0x80000

L3 Cache size in bytes.

l3cache-ways**Type**

int

Default value

0x10

L3 Cache number of ways (sets are implicit from size).

log2_trace_buffer_alignment**Type**

int

Default value

0x6

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

memory_tagging_support_level**Type**

int

Default value

0x3

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

MPAM Whether MPAMIDR_EL1.HAS_ALTSP bit is set or clear.

mpamidr_has_force_ns**Type**

int

Default value

0x0

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

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

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

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

Number of ACP ports.

num_nodes**Type**

int

Default value

0x1

Number of transport nodes. Zero implies direct-connect configuration.

pmu-num_counters**Type**

int

Default value

0x1f

Number of PMU counters implemented.

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

tcr_txsz_undersize_should_fault**Type**

bool

Default value

0x0

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.74 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 3-487: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARM CortexX925CT

This model has the following Iris instances:

Table 3-488: ARM CortexX925CT Iris instances

InstanceName	ComponentName
ARM CortexX925CT	Cluster_ARM_Cortex-X925
ARM CortexX925CT.AMU	PVBusLogger
ARM CortexX925CT.AMU.mapper	PVBusMapper
ARM CortexX925CT.DAP	PVBusLogger
ARM CortexX925CT.DAP.mapper	PVBusMapper
ARM CortexX925CT.DSU	DSU
ARM CortexX925CT.DSU.PPU_cluster	PPUv1
ARM CortexX925CT.DSU.PPU_cluster.busslave	PVBusSlave
ARM CortexX925CT.DSU.PPU_core0	PPUv1
ARM CortexX925CT.DSU.PPU_core0.busslave	PVBusSlave
ARM CortexX925CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM CortexX925CT.DSU.mpam_busslave	PVBusSlave
ARM CortexX925CT.DSU.shared_cache	PVCache
ARM CortexX925CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARM CortexX925CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARM CortexX925CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARM CortexX925CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARM CortexX925CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARM CortexX925CT.DSU.utility_slave[0]	PVBusSlave
ARM CortexX925CT.MMAP	PVBusLogger
ARM CortexX925CT.MMAP.mapper	PVBusMapper
ARM CortexX925CT.RAS	PVBusLogger
ARM CortexX925CT.RAS.mapper	PVBusMapper
ARM CortexX925CT.cpu0	ARM_Cortex-X925
ARM CortexX925CT.cpu0.UTLB	TLB
ARM CortexX925CT.cpu0.dtlb	TlbCadi
ARM CortexX925CT.cpu0.l1dcache	PVCache
ARM CortexX925CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM CortexX925CT.cpu0.l1icache	PVCache
ARM CortexX925CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARM CortexX925CT.cpu0.l2cache	PVCache
ARM CortexX925CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARM CortexX925CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARM CortexX925CT.ext_bus	PVBusLogger
ARM CortexX925CT.ext_bus.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexX925CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-489: ARMCortexX925CT MTI instances

InstanceName	ComponentName
ARMCortexX925CT	ARMv8Cluster
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
ARMCortexX925CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX925CT.ext_bus	PVBusLogger

InstanceName	ComponentName
ARMCortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexX925CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder

ARMCortexX925CT contains the following CADI targets:

- ARM_Cortex-X925
- Cluster_ARM_Cortex-X925
- PVCache
- TlbCadi

Ports for ARMCortexX925CT

Table 3-490: 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_pcsmpchannel	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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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 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.
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.

Name	Protocol	Type	Description
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.
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).

Name	Protocol	Type	Description
vsei[14]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM CortexX925CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

Value of CPU Configuration Register.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

bus_type**Type**

int

Default value

0x0

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

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 PPUs 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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX . CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX . CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX . CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

ecv_support_level**Type**

int

Default value

0x2

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

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

Number of claim tags.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

ext_abort_so_write_ras_type**Type**

int

Default value

0x2

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

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

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

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

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

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

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

If true, an additional AXI peripheral port is configured.

has_statistical_profiling**Type**

bool

Default value

0x1

Whether Statistical Based Profiling is implemented (FEAT_SPE).

has_v8_7_spe_inverted_filtering**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the inverting filtering by events.

has_v8_7_spe_previous_branch_target**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the optional branch target feature is implemented.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

l3cache-size**Type**

int

Default value

0x80000

L3 Cache size in bytes.

l3cache-ways**Type**

int

Default value

0x10

L3 Cache number of ways (sets are implicit from size).

log2_trace_buffer_alignment**Type**

int

Default value

0x6

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

memory_tagging_support_level**Type**

int

Default value

0x3

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

MPAM Whether MPAMIDR_EL1.HAS_ALTSP bit is set or clear.

mpamidr_has_force_ns**Type**

int

Default value

0x0

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

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

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

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

Number of ACP ports.

num_nodes**Type**

int

Default value

0x1

Number of transport nodes. Zero implies direct-connect configuration.

pmu-num_counters**Type**

int

Default value

0x1f

Number of PMU counters implemented.

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

tcr_txsz_undersize_should_fault**Type**

bool

Default value

0x0

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.75 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 3-491: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMNeoverseE1CT

This model has the following Iris instances:

Table 3-492: ARMNeoverseE1CT Iris instances

InstanceName	ComponentName
ARMNeoverseE1CT	Cluster_ARM_Neoverse-E1
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
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	TlbCadi
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

InstanceName	ComponentName
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

This model has the following MTI trace components:

Table 3-493: ARMNeoverseE1CT MTI instances

InstanceName	ComponentName
ARMNeoverseE1CT	ARMv8Cluster
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
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

InstanceName	ComponentName
ARMNeoverseE1CT.cpu0.thread1.UTLB	TLB
ARMNeoverseE1CT.ext_bus	PVBusLogger
ARMNeoverseE1CT.ext_bus.mapper	PVBusMapper
ARMNeoverseE1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

ARMNeoverseE1CT contains the following CADI targets:

- ARM_Neoverse-E1
- Cluster_ARM_Neoverse-E1
- PVCache
- TlbCadi

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 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](#).

Ports for ARMNeoverseE1CT

Table 3-494: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port
<code>AENDMP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>ASTARTMP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>broadcastatomic</code>	Signal	Slave	CHI defined pins.
<code>broadcastcachemaint</code>	Signal	Slave	ACE defined pins.
<code>broadcastouter</code>	Signal	Slave	ACE defined pins.
<code>broadcastpersist</code>	Signal	Slave	CHI defined pins.
<code>cfgend[16]</code>	Signal	Slave	This signal is for EE bit initialisation.
<code>cfgsdisable</code>	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
<code>cfgte[16]</code>	Signal	Slave	This signal provides default exception handling state.
<code>clk_in</code>	ClockSignal	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the core executes instructions.
<code>clusterid</code>	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
<code>clusterpmuirq</code>	Signal	Master	DynamlQ pmu irq
<code>CNTHPIRQ[16]</code>	Signal	Master	Timer signals to SOC.
<code>CNTHVIRQ[16]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPNSIRQ[16]</code>	Signal	Master	Timer signals to SOC.
<code>CNTPSIRQ[16]</code>	Signal	Master	Timer signals to SOC.
<code>cntvalueb</code>	CounterInterface	Slave	Interface to SoC level counter module.
<code>CNTVIRQ[16]</code>	Signal	Master	Timer signals to SOC.
<code>commirq[16]</code>	Signal	Master	Interrupt signal from debug communications channel.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

cluster_patch_level**Type**

int

Default value

0x0

Cosmetic change to patch number in CLUSTERIDR. Corresponds to the Y in rXpY.

cluster_revision_number**Type**

int

Default value

0x0

Cosmetic change to revision number in CLUSTERIDR, Corresponds to the X in rXpY.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.enable_single_thread_at_reset**Type**

bool

Default value

0x0

Enable single thread after reset and keep other thread in reset.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.threadY.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.threadY.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.threadY.MPIDR-override**Type**

int

Default value

0x0

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

Value of RVBAR_ELx register.

cpuX.threadY.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

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

If true, Accelerator Coherency Port is configured.

has_dot_product**Type**

int

Default value

0x2

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

has_peripheral_port**Type**

bool

Default value

0x0

If true, additional AXI peripheral port is configured.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

`icache-state_modelled`

Type

bool

Default value

0x0

Set whether I-cache has stateful implementation.

`l3cache-hit_latency`

Type

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end

Type

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start

Type

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency

Type

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency

Type

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.76 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 3-495: IP revisions support

Revision	Quality level
r4p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMNeoverseN1CT

This model has the following Iris instances:

Table 3-496: 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

InstanceName	ComponentName
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
ARMNeoverseN1CT.cpu0	ARM_Neoverse-N1
ARMNeoverseN1CT.cpu0.UTLB	TLB
ARMNeoverseN1CT.cpu0.dtlb	TlbCadi
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

This model has the following MTI trace components:

Table 3-497: ARMNeoverseN1CT MTI instances

InstanceName	ComponentName
ARMNeoverseN1CT	ARMv8Cluster
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

InstanceName	ComponentName
ARMNeoverseN1CT.MMAP	PVBusLogger
ARMNeoverseN1CT.MMAP.mapper	PVBusMapper
ARMNeoverseN1CT.RAS	PVBusLogger
ARMNeoverseN1CT.RAS.mapper	PVBusMapper
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

ARMNeoverseN1CT contains the following CADI targets:

- ARM_Neoverse-N1
- Cluster_ARM_Neoverse-N1
- PVCache
- TlbCadi

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 `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 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](#).

Ports for ARMNeoverseN1CT

Table 3-498: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>AENDMP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>ASTARTMP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>broadcastatomic</code>	Signal	Slave	CHI defined pins.
<code>broadcastcachemaint</code>	Signal	Slave	ACE defined pins.
<code>broadcastouter</code>	Signal	Slave	ACE defined pins.
<code>broadcastpersist</code>	Signal	Slave	CHI defined pins.
<code>cfgend[4]</code>	Signal	Slave	This signal is for EE bit initialisation.
<code>cfgsdisable</code>	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
<code>cfgte[4]</code>	Signal	Slave	This signal provides default exception handling state.
<code>clk_in</code>	ClockSignal	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
<code>clusterid</code>	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
<code>clusterpmuirq</code>	Signal	Master	DynamlQ pmu irq
<code>CNTHPIRQ[4]</code>	Signal	Master	Timer signals to SOC.

Name	Protocol	Type	Description
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.
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 ARMNeoverseN1CT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

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

Number of cores per cluster.

cpi_div**Type**

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`cpuX.l2cache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

`dcache-read_access_latency`

Type

int

Default value

0x0

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

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

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations

Type

bool

Default value

0x1

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

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_write_is_sync

Type

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_so_read_is_sync

Type

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_write_is_sync

Type

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

If true, Accelerator Coherency Port is configured.

has_coherent_icache**Type**

bool

Default value

0x0

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

If true, additional AXI peripheral port is configured.

has_statistical_profiling**Type**

bool

Default value

0x1

Whether Statistical Based Profiling is implemented (FEAT_SPE).

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

Set whether I-cache has stateful implementation.

`l3cache-hit_latency`

Type

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end**Type**

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.77 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 3-499: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMNeoverseN2CT

This model has the following Iris instances:

Table 3-500: 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

InstanceName	ComponentName
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
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.dtlb	TlbCadi
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
ARMNeoverseN2CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-501: ARMNeoverseN2CT MTI instances

InstanceName	ComponentName
ARMNeoverseN2CT	ARMv8Cluster
ARMNeoverseN2CT.AMU	PVBusLogger
ARMNeoverseN2CT.AMU.mapper	PVBusMapper
ARMNeoverseN2CT.DAP	PVBusLogger
ARMNeoverseN2CT.DAP.mapper	PVBusMapper
ARMNeoverseN2CT.DSU	DSU
ARMNeoverseN2CT.DSU.PPU_cluster	PPUv1

InstanceName	ComponentName
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
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

ARMNeoverseN2CT contains the following CADI targets:

- ARM_Neoverse-N2
- Cluster_ARM_Neoverse-N2
- PVCache
- TlbCadi

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
- `COREINSTRET`, `COREINSTRUN`, 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 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](#).

Ports for ARMNeoverseN2CT

Table 3-502: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<code>AEND0MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND1MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND2MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>AEND3MP</code>	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
<code>ASTART0MP</code>	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
<code>ASTART1MP</code>	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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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_pcsmp_channel[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

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

core_power_on_by_default**Type**

bool

Default value

0x0

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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`

Type

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Number of claim tags.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

ext_abort_device_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_read_ras_type**Type**

int

Default value

0x0

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

Synchronous reporting of device-nGnRnE write external aborts.

force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

If true, Accelerator Coherency Port is configured.

has_coherent_icache**Type**

bool

Default value

0x0

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

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

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

If true, an additional AXI peripheral port is configured.

has_rndr**Type**

int

Default value

0x1

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

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`

Type

int

Default value

0x0

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

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

Set whether I-cache has stateful implementation.

l3cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

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

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

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

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

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

MPAM Maximum PARTID Supported.

mpmm_accumulator_multiplier**Type**

int

Default value

0x1

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

Register core as ON state to cluster with simulation reset.

periph_address_end**Type**

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start**Type**

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

rndr_rndrrs_seed**Type**

int

Default value

0x0

Initial seed for random engine used in RNDR register.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.78 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 3-503: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMNeoverseN3CT

This model has the following Iris instances:

Table 3-504: 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
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.dtlb	TlbCadi
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
ARMNeoverseN3CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-505: ARMNeoverseN3CT MTI instances

InstanceName	ComponentName
ARMNeoverseN3CT	ARMv8Cluster
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
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

ARMNeoverseN3CT contains the following CADI targets:

- ARM_Neoverse-N3
- Cluster_ARM_Neoverse-N3

- PVCache
- TlbCadi

Ports for ARMNeoverseN3CT

Table 3-506: Ports

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AENDOMP	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).
ASTARTOMP	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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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.

Name	Protocol	Type	Description
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 PPUs 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 PPUs 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.

Name	Protocol	Type	Description
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.
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

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

Value of CPU Configuration Register.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

bus_type**Type**

int

Default value

0x0

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

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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations**Type**

bool

Default value

0x1

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

Number of claim tags.

ete.ETE_REVISION**Type**

int

Default value

0x1

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x1

Return stack depth.

ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

ete.TSMARK**Type**

bool

Default value

0x1

Whether timestamp markers are supported.

force_mte_tag_access_razwi_and_ignore_tag_checks**Type**

bool

Default value

0x0

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

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

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

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

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

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

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

If true, an additional AXI peripheral port is configured.

has_rndr_trap

Type

int

Default value

0x2

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

Whether Statistical Based Profiling is implemented (FEAT_SPE).

icache-hit_latency

Type

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

memory_tagging_support_level**Type**

int

Default value

0x3

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

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.

pmu-num_counters**Type**

int

Default value

0x6

Number of PMU counters implemented.

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.79 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 3-507: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMNeoverseV1CT

This model has the following Iris instances:

Table 3-508: ARMNeoverseV1CT Iris instances

InstanceName	ComponentName
ARMNeoverseV1CT	Cluster_ARM_Neoverse-V1
ARMNeoverseV1CT.AMU	PVBusLogger
ARMNeoverseV1CT.AMU.mapper	PVBusMapper
ARMNeoverseV1CT.DAP	PVBusLogger

InstanceName	ComponentName
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.dtlb	TlbCadi
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.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-509: ARMNeoverseV1CT MTI instances

InstanceName	ComponentName
ARMNeoverseV1CT	ARMv8Cluster
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

InstanceName	ComponentName
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
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 contains the following CADI targets:

- ARM_Neoverse-V1
- Cluster_ARM_Neoverse-V1
- PVCache
- TlbCadi

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 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](#).

Ports for ARMNeoverseV1CT

Table 3-510: 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 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.
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.

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[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 ARMNeoverseV1CT

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x1

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x1

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

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

Number of cores per cluster.

cpi_div

Type

int

Default value

0x1

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul

Type

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND

Type

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.VINITHI**Type**

bool

Default value

0x0

Reset value of SCTLR.V.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

`cpuX.l2cache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

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

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

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

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

enable_simulation_performance_optimizations

Type

bool

Default value

0x1

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

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 cherrypicked 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

Synchronous reporting of device-nGnRE read external aborts.

ext_abort_device_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRE write external aborts.

ext_abort_so_read_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE read external aborts.

ext_abort_so_write_is_sync**Type**

bool

Default value

0x0

Synchronous reporting of device-nGnRnE write external aborts.

force_zero_PSTATE_PAN**Type**

bool

Default value

0x0

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

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

If true, Accelerator Coherency Port is configured.

has_coherent_icache**Type**

bool

Default value

0x1

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

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

If true, additional AXI peripheral port is configured.

has_rndr**Type**

int

Default value

0x1

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

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

`icache-state_modelled`

Type

bool

Default value

0x0

Set whether I-cache has stateful implementation.

`l3cache-hit_latency`

Type

int

Default value

0x0

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

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

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

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

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

L3 Cache size in bytes.

l3cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

Register core as ON state to cluster with simulation reset.

periph_address_end

Type

int

Default value

0x0

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

periph_address_start

Type

int

Default value

0x0

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

ptw_latency

Type

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

tlb_latency

Type

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled

Type
bool

Default value
0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.80 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 3-511: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMNeoverseV2CT

This model has the following Iris instances:

Table 3-512: 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
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.dtlb	TlbCadi
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
ARMNeoverseV2CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-513: ARMNeoverseV2CT MTI instances

InstanceName	ComponentName
ARMNeoverseV2CT	ARMv8Cluster
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
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

ARMNeoverseV2CT contains the following CADI targets:

- ARM_Neoverse-V2
- Cluster_ARM_Neoverse-V2
- PVCache

- TlbCadi

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](#).

The following models also exist, with a fixed number of cores per cluster:

- ARMNeoverseV2x1CT.

Ports for ARMNeoverseV2CT

Table 3-514: 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[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
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
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

Name	Protocol	Type	Description
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 PPUs 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 PPUs 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.

Name	Protocol	Type	Description
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.
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

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT

Type

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT**Type**

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER**Type**

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST**Type**

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Value of CPU Configuration Register.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

bus_type**Type**

int

Default value

0x0

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

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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.enable_trace_special_hlt_imm16**Type**

bool

Default value

0x0

Enable usage of parameter trace_special_hlt_imm16.

cpuX.force-fpsid**Type**

bool

Default value

0x1

Override the FPSID value.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

`dcache-snoop_data_transfer_latency`

Type

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`

Type

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

`enable_simulation_performance_optimizations`**Type**

bool

Default value

0x1

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

Number of claim tags.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.PIDR_REVAND**Type**

int

Default value

0x0

TRCPIDR REVAND value.

ete.PIDR_REVISION**Type**

int

Default value

0x0

TRCPIDR REVISION value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.REVISION**Type**

int

Default value

0x0

TRCIDR1 revision value.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

ext_abort_normal_noncacheable_read_is_sync**Type**

bool

Default value

0x1

Synchronous reporting of normal noncacheable-read external aborts.

ext_abort_so_write_ras_type**Type**

int

Default value

0x2

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

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

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

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 MPAMO_EL1, MPAM1_EL1, MPAM2_EL2, MPAM3_EL3, MPAMHCR_EL2.

has_coherent_icache

Type

bool

Default value

0x1

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

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

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

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

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

If true, an additional AXI peripheral port is configured.

has_rndr**Type**

int

Default value

0x1

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

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

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

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

Enable simulation of instruction cache prefetching. This is only used when icache-state_modelled=true.

icache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

memory_tagging_support_level**Type**

int

Default value

0x3

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

MPAM Maximum PARTID Supported.

mpam_max_vpmr**Type**

int

Default value

0x7

MPAM Maximum VPMR Supported.

mpmm_accumulator_multiplier**Type**

int

Default value

0x1

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

Number of PMU counters implemented.

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

rndr_rndrrs_seed**Type**

int

Default value

0x0

Initial seed for random engine used in RNDR register.

stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT_PAAuth 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_PAAuth traps.

walk_cache_latency

Type

int

Default value

0x0

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.81 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 3-515: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Changes in 11.27.17

Parameters added:

- brbe_log2_num_records
- has_coherent_icache

Iris and MTI instances for ARMNeoverseV3AECT

This model has the following Iris instances:

Table 3-516: 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

InstanceName	ComponentName
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.dtlb	TlbCadi
ARMNeoverseV3AECT.cpu0.l1dcache	PVCache
ARMNeoverseV3AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
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
ARMNeoverseV3AECT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-517: ARMNeoverseV3AECT MTI instances

InstanceName	ComponentName
ARMNeoverseV3AECT	ARMv8Cluster
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

InstanceName	ComponentName
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
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

ARMNeoverseV3AECT contains the following CADI targets:

- ARM_Neoverse-V3AE
- Cluster_ARM_Neoverse-V3AE
- PVCache
- TlbCadi

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.

Ports for ARMNeoverseV3AECT

Table 3-518: 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_pcsmp_channel	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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

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[1]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARMNeoverseV3AECT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMPO input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC**Type**

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled

Type

int

Default value

0x1

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

Value of CPU Configuration Register.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

brbe_log2_num_records**Type**

int

Default value

0x5

Log2 of number of BRB records supported. 3 -> 8 records, ... 6 -> 64 records.

bus_type**Type**

int

Default value

0x0

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

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 PPUs 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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.crypto_aes**Type**

int

Default value

0x2

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

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

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

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

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

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

dcache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

dcache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

ecv_support_level**Type**

int

Default value

0x2

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

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

Number of claim tags.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCRSRTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCRSR.TA value for a forcibly traced exception.

ext_abort_so_write_ras_type**Type**

int

Default value

0x2

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

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

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

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 MPAMO_EL1, MPAM1_EL1, MPAM2_EL2, MPAM3_EL3, MPAMHCR_EL2.

has_coherent_icache

Type

bool

Default value

0x1

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

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

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

If true, an additional AXI peripheral port is configured.

has_rndr

Type

int

Default value

0x1

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

Whether Statistical Based Profiling is implemented (FEAT_SPE).

has_v8_7_spe_inverted_filtering**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the inverting filtering by events.

has_v8_7_spe_previous_branch_target**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the optional branch target feature is implemented.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

l3cache-ways**Type**

int

Default value

0x10

L3 Cache number of ways (sets are implicit from size).

log2_trace_buffer_alignment**Type**

int

Default value

0x6

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

memory_tagging_support_level**Type**

int

Default value

0x3

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

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

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

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

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

Number of ACP ports.

ptw_latency**Type**

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

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

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

Initial seed for random engine used in RNDR register.

stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

tcr_txsz_undersize_should_fault**Type**

bool

Default value

0x0

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT_PAAuth 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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.82 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 3-519: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Changes in 11.27.17

Parameters added:

- `brbe_log2_num_records`
- `has_coherent_icache`

Iris and MTI instances for ARMNeoverseV3CT

This model has the following Iris instances:

Table 3-520: 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
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.dtlb	TlbCadi

InstanceName	ComponentName
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
ARMNeoverseV3CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Table 3-521: ARMNeoverseV3CT MTI instances

InstanceName	ComponentName
ARMNeoverseV3CT	ARMv8Cluster
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
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

InstanceName	ComponentName
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

ARMNeoverseV3CT contains the following CADI targets:

- ARM_Neoverse-V3
- Cluster_ARM_Neoverse-V3
- PVCache
- TlbCadi

Ports for ARMNeoverseV3CT

Table 3-522: 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.

Name	Protocol	Type	Description
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 port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
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_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.
dbgprupreq[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 ARMNeoverseV3CT

AEND0_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

AEND1_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

AEND2_DEFAULT

Type

int

Default value

0x0

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

AEND3_DEFAULT**Type**

int

Default value

0x0

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

ASTART0_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

ASTART1_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

ASTART2_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

ASTART3_DEFAULT**Type**

int

Default value

0x0

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

BROADCASTATOMIC

Type

bool

Default value

0x1

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

BROADCASTCACHEMAINT

Type

bool

Default value

0x0

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

BROADCASTOUTER

Type

bool

Default value

0x0

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

BROADCASTPERSIST

Type

bool

Default value

0x1

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

Processor cluster ID value.

CMO_broadcast_when_cache_state_modelling_disabled**Type**

int

Default value

0x1

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

Value of CPU Configuration Register.

GICDISABLE**Type**

bool

Default value

0x1

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

Number of cores per cluster.

brbe_log2_num_records**Type**

int

Default value

0x5

Log2 of number of BRB records supported. 3 -> 8 records, ... 6 -> 64 records.

bus_type**Type**

int

Default value

0x0

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

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

Divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

Multiplier for calculating CPI (Cycles Per Instruction).

cpuX.CFGEND**Type**

bool

Default value

0x0

Endianness configuration at reset. 0, little endian. 1, big endian.

cpuX.CFGTE**Type**

bool

Default value

0x0

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

cpuX.CRYPTODISABLE**Type**

bool

Default value

0x0

Disable cryptographic features.

cpuX.RVBARADDR**Type**

int

Default value

0x0

Value of RVBAR_ELx register.

cpuX.crypto_aes**Type**

int

Default value

0x2

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

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

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

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

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

Enable usage of parameter trace_special_hlt_imm16.

cpuX.l2cache-hit_latency**Type**

int

Default value

0x0

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

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

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

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

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

L2 Cache size in bytes.

cpuX.l2cache-snoop_data_transfer_latency**Type**

int

Default value

0x0

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

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

L2 Cache number of ways (sets are implicit from size).

cpuX.l2cache-write_access_latency**Type**

int

Default value

0x0

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

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

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

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

cpuX.semihosting-A32_HLT**Type**

int

Default value

0xf000

A32 HLT number for semihosting calls.

cpuX.semihosting-A64_HLT**Type**

int

Default value

0xf000

A64 HLT number for semihosting calls.

cpuX.semihosting-ARM_SVC**Type**

int

Default value

0x123456

A32 SVC number for semihosting calls.

cpuX.semihosting-T32_HLT**Type**

int

Default value

0x3c

T32 HLT number for semihosting calls.

cpuX.semihosting-Thumb_SVC**Type**

int

Default value

0xab

T32 SVC number for semihosting calls.

cpuX.semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting calls.

cpuX.semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

cpuX.semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC/HLT traps.

cpuX.semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

cpuX.semihosting-heap_limit**Type**

int

Default value

0xf000000

Virtual address of top of heap.

cpuX.semihosting-stack_base**Type**

int

Default value

0x10000000

Virtual address of base of descending stack.

cpuX.semihosting-stack_limit**Type**

int

Default value

0xf000000

Virtual address of stack limit.

cpuX.trace_special_hlt_imm16**Type**

int

Default value

0xf000

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

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

Set whether the model has VFP support.

dcache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of data cache prefetching. This is only used when dcache-state_modelled=true.

dcache-read_access_latency**Type**

int

Default value

0x0

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

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

L1 D-Cache size in bytes.

`dcache-snoop_data_transfer_latency`**Type**

int

Default value

0x0

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

Set whether D-cache has stateful implementation.

`dcache-write_access_latency`**Type**

int

Default value

0x0

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

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

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

Enable DynamIQ diagnostic messages.

ecv_support_level**Type**

int

Default value

0x2

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

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

Number of claim tags.

ete.MAX_INST_PER_Q**Type**

int

Default value

0x1

Maximum limit for the number of instructions implied by a Q element.

ete.NumberOfRSPairs**Type**

int

Default value

0x8

Number of resource selector pairs.

ete.PIDR_CM0D**Type**

int

Default value

0x0

TRCPIDR CM0D value.

ete.Q_CADENCE**Type**

int

Default value

0x1

Number of instruction blocks traced between two Q elements.

ete.RES0_STATEFUL**Type**

bool

Default value

0x0

Whether RES0 bits are stateful or RAZ/WI.

ete.RETSTACK**Type**

int

Default value

0x3

Return stack depth.

ete.SIM_OVERFLOW_GRANULARITY**Type**

int

Default value

0x64

Number of instruction blocks in each granule, for simulated overflow.

ete.SIM_OVERFLOW_PERCENTAGE**Type**

int

Default value

0x0

Percentage of instruction blocks lost in each granule, for simulated overflow.

ete.SOURCE_ADDRESS**Type**

bool

Default value

0x0

Allow generation of source address elements.

ete.TRACE_OUTPUT**Type**

string

Default value

""

File to which to write trace byte stream.

ete.TRCSRSTA_FORCED_EXCEP**Type**

bool

Default value

0x0

TRCSRSTA value for a forcibly traced exception.

ext_abort_so_write_ras_type**Type**

int

Default value

0x2

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

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

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

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 MPAMO_EL1, MPAM1_EL1, MPAM2_EL2, MPAM3_EL3, MPAMHCR_EL2.

has_coherent_icache

Type

bool

Default value

0x1

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

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

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

If true, an additional AXI peripheral port is configured.

has_rndr**Type**

int

Default value

0x1

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

Whether Statistical Based Profiling is implemented (FEAT_SPE).

has_v8_7_spe_inverted_filtering**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the inverting filtering by events.

has_v8_7_spe_previous_branch_target**Type**

bool

Default value

0x0

Where FEAT_SPEv1p2 is implemented, whether the optional branch target feature is implemented.

icache-hit_latency**Type**

int

Default value

0x0

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

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

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

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

`icache-read_access_latency`**Type**

int

Default value

0x0

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

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

L1 I-Cache size in bytes.

icache-state_modelled**Type**

bool

Default value

0x0

Set whether I-cache has stateful implementation.

instruction_tlb_size**Type**

int

Default value

0x0

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

l3cache-ways**Type**

int

Default value

0x10

L3 Cache number of ways (sets are implicit from size).

log2_trace_buffer_alignment**Type**

int

Default value

0x6

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

memory_tagging_support_level**Type**

int

Default value

0x3

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

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

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

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

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

Number of ACP ports.

ptw_latency

Type

int

Default value

0x0

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

reported_patch_level

Type

int

Default value

0xffffffffffffffff

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

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

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

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

Initial seed for random engine used in RNDR register.

stage12_tlb_size**Type**

int

Default value

0x80

Number of stage1+2 tlb entries.

tcr_txsz_undersize_should_fault**Type**

bool

Default value

0x0

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

tlb_latency**Type**

int

Default value

0x0

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

tlbi_stall_enabled**Type**

bool

Default value

0x0

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

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

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.83 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 3-523: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMSC000CT

This model has the following Iris instances:

Table 3-524: 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 3-525: 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

ARMSC000CT contains the following CADI targets:

- ARM_SC000

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.

Ports for ARMSC000CT

Table 3-526: 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.
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 ARMSC000CT

BIGENDINIT

Type

bool

Default value

0x0

Initialize processor to big endian mode.

BKPT**Type**

int

Default value

0x4

Number of breakpoint unit comparators implemented.

DBG**Type**

bool

Default value

0x1

Set whether debug extensions are implemented.

IOP**Type**

bool

Default value

0x0

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

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n].

NUM_IRQ**Type**

int

Default value

0x20

Number of user interrupts.

NUM_MPU_REGION

Type

int

Default value

0x0

Number of MPU regions.

SYST

Type

bool

Default value

0x1

Enable support for SysTick timer functionality.

USER

Type

bool

Default value

0x1

Enable support for Unprivileged/Privileged Extension.

VTOR

Type

bool

Default value

0x1

Include Vector Table Offset Register.

WIC

Type

bool

Default value

0x1

Include support for WIC-mode deep sleep.

WPT**Type**

int

Default value

0x2

Number of watchpoint unit comparators implemented.

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit**Type**

int

Default value

0x20700000

Virtual address of top of heap.

semihosting-stack_base**Type**

int

Default value

0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit**Type**

int

Default value

0x20700000

Virtual address of stack limit.

3.5.84 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 3-527: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for ARMSC300CT

This model has the following Iris instances:

Table 3-528: 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 3-529: 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

ARMSC300CT contains the following CADI targets:

- ARM_SC300

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:

Table 3-530: ITM_PACKET_TYPE field values that 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.

Ports for ARMSC300CT

Table 3-531: 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.
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.

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.
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

Enable bitbanding.

BIGENDINIT

Type

bool

Default value

0x0

Initialize processor to big endian mode.

DBGLVL

Type

int

Default value

0x3

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

Number of bits of interrupt priority.

NUM_IRQ**Type**

int

Default value

0x10

Number of user interrupts.

NUM_MPU_REGION**Type**

int

Default value

0x8

Number of MPU regions.

TRACE_LVL**Type**

int

Default value

0x1

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

Include support for WIC-mode deep sleep.

cpi_div**Type**

int

Default value

0x1

divider for calculating CPI (Cycles Per Instruction).

cpi_mul**Type**

int

Default value

0x1

multiplier for calculating CPI (Cycles Per Instruction).

master_id**Type**

int

Default value

0x0

Master ID presented in bus transactions.

min_sync_level**Type**

int

Default value

0x0

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

reported_patch_level**Type**

int

Default value

0xffffffffffffffff

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

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

T32 SVC number for semihosting.

semihosting-cmd_line**Type**

string

Default value

""

Command line available to semihosting SVC calls.

semihosting-cwd**Type**

string

Default value

""

Base directory for semihosting file access.

semihosting-enable**Type**

bool

Default value

0x1

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

semihosting-heap_base**Type**

int

Default value

0x0

Virtual address of heap base.

semihosting-heap_limit

Type
int

Default value
0x20700000

Virtual address of top of heap.

semihosting-stack_base

Type
int

Default value
0x20800000

Virtual address of base of descending stack.

semihosting-stack_limit

Type
int

Default value
0x20700000

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 3-532: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for D71

This model has the following Iris instances:

Table 3-533: 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 3-534: D71 MTI instances

InstanceName	ComponentName
D71	D71
D71.apb_slave_adu	PVBusSlave
D71.apb_slave_dpu	PVBusSlave

D71 contains the following CADI targets:

- D71

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.

Ports for D71

Table 3-535: 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
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)

Name	Protocol	Type	Description
pvbustbu_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

Non-protected NSAID for ADU transactions.

adu_nprot_s2_sid

Type

int

Default value

0x2

Stage 2 non-protected StreamID for ADU transactions.

adu_prot_nsaid

Type

int

Default value

0x1

Protected NSAID for ADU transactions.

adu_prot_s2_sid

Type

int

Default value

0x5

Stage 2 protected StreamID for ADU transactions.

adu_rd_s1_sid

Type

int

Default value

0xa

Stage 1 StreamID for ADU DMA read layer.

adu_wr_s1_sid**Type**

int

Default value

0xb

Stage 1 StreamID for ADU AES write-back layer.

display_split_en**Type**

int

Default value

0x0

Display split enabled or not.

force_frame_rate_0**Type**

int

Default value

0x0

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

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

Stage 1 StreamID for LPU0 read layer 0.

lpu0_l1_s1_sid**Type**

int

Default value

0x1

Stage 1 StreamID for LPU0 read layer 1.

lpu0_l2_s1_sid**Type**

int

Default value

0x2

Stage 1 StreamID for LPU0 read layer 2.

lpu0_l3_s1_sid**Type**

int

Default value

0x3

Stage 1 StreamID for LPU0 read layer 3.

lpu0_nprot_nsa_id**Type**

int

Default value

0x0

Non-protected NSAID for LPU0 transactions.

lpu0_nprot_s2_sid**Type**

int

Default value

0x0

Stage 2 non-protected StreamID for LPU0 transactions.

lpu0_prot_nsaid

Type

int

Default value

0x1

Protected NSAID for LPU0 transactions.

lpu0_prot_s2_sid

Type

int

Default value

0x3

Stage 2 protected StreamID for LPU0 transactions.

lpu0_wr_s1_sid

Type

int

Default value

0x8

Stage 1 StreamID for LPU0 write-back layer.

lpu1_l0_s1_sid

Type

int

Default value

0x4

Stage 1 StreamID for LPU1 read layer 0.

lpu1_l1_s1_sid

Type

int

Default value

0x5

Stage 1 StreamID for LPU1 read layer 1.

lpu1_12_s1_sid**Type**

int

Default value

0x6

Stage 1 StreamID for LPU1 read layer 2.

lpu1_13_s1_sid**Type**

int

Default value

0x7

Stage 1 StreamID for LPU1 read layer 3.

lpu1_nprot_nsaaid**Type**

int

Default value

0x0

Non-protected NSAID for LPU1 transactions.

lpu1_nprot_s2_sid**Type**

int

Default value

0x1

Stage 2 non-protected StreamID for LPU1 transactions.

lpu1_prot_nsaaid**Type**

int

Default value

0x1

Protected NSAID for LPU1 transactions.

lpu1_prot_s2_sid

Type
int

Default value
0x4

Stage 2 protected StreamID for LPU1 transactions.

lpu1_wr_s1_sid

Type
int

Default value
0x9

Stage 1 StreamID for LPU1 write-back layer.

num_rich_layers

Type
int

Default value
0x2

Number of Rich layers in each Layer Processing Unit.

3.6.2 DP500

ARM DP500 Display Processor. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-536: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for DP500

This model has the following Iris instances:

Table 3-537: DP500 Iris instances

InstanceName	ComponentName
DP500	DP500
DP500.busmaster	PVBusMaster

InstanceName	ComponentName
DP500.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-538: DP500 MTI instances

InstanceName	ComponentName
DP500	DP500
DP500.busmaster	PVBusMaster
DP500.busslave	PVBusSlave

DP500 contains the following CADI targets:

- DP500

About DP500

This model has basic support for the display and scaling engines. Connect it to a visualization component to view LCD output. This is the single display configuration of DP500. For the dual display configuration, use DP500x2.

It passes tests as part of a booting Android kernel and running under the control of the official DP500 drivers.

It provides the following functionality:

- All RGB and YUV format parsing.
- Color adjustment in Display Engine (DE).
- Nearest neighbor scaling.
- All layers.
- Alpha blending.
- Memory writeback.
- Inverse gamma adjustment.
- Basic layer (overlay) and register security semantics.

Limitations

- No support for polyphase scaling algorithm. Falls back to nearest neighbor when configured to do so.
- No support for 3D or interlaced video.
- No support for image enhancing functionality.
- No colorspace conversion support.
- No support for two plane YUV memory writeback.

Ports for DP500

Table 3-539: 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.
intr	Signal	Master	Interrupt signal.
intr_se	Signal	Master	Interrupt signal from scaling engine.
pvbus_m	PVBus	Master	Bus for processor 0.
pvbus_s	PVBus	Slave	Slave port for register access.
reset_signal	Signal	Slave	Reset signal.

Parameters for DP500

disable_snooping_dma

Type

bool

Default value

0x0

Disable DMA snooping.

dual_mode

Type

bool

Default value

0x0

Whether 2 display mode is supported.

force_frame_rate

Type

int

Default value

0x0

If 0 - the input clock is used as PXLCLK, if >0 then the model ensures the screen display is refreshed n times per simulated second.

3.6.3 DP500x2

ARM DP500 Display Processor x2. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-540: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for DP500x2

This model has the following Iris instances:

Table 3-541: DP500x2 Iris instances

InstanceName	ComponentName
DP500x2	DP500x2
DP500x2.display_core_0	DP500
DP500x2.display_core_0.busmaster	PVBusMaster
DP500x2.display_core_0.busslave	PVBusSlave
DP500x2.display_core_1	DP500
DP500x2.display_core_1.busmaster	PVBusMaster
DP500x2.display_core_1.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-542: DP500x2 MTI instances

InstanceName	ComponentName
DP500x2.display_core_0	DP500
DP500x2.display_core_0.busmaster	PVBusMaster
DP500x2.display_core_0.busslave	PVBusSlave
DP500x2.display_core_1	DP500
DP500x2.display_core_1.busmaster	PVBusMaster
DP500x2.display_core_1.busslave	PVBusSlave

DP500x2 contains the following CADI targets:

- DP500
- DP500x2

About DP500x2

This component is a model of the dual display configuration of the DP500 Display Processor, with basic support for the display and scaling engines. Connect it to a visualization component to view LCD output.

It provides the following functionality:

- All RGB and YUV format parsing.
- Color adjustment in Display Engine (DE).
- Nearest neighbor scaling.
- All layers.
- Alpha blending.
- Memory writeback.
- Inverse gamma adjustment.
- Basic layer (overlay) and register security semantics.

Limitations

- No support for polyphase scaling algorithm. Falls back to nearest neighbor when configured to do so.
- No support for 3D or interlaced video.
- No support for image enhancing functionality.
- No colorspace conversion support.
- No support for two plane YUV memory writeback.

Ports for DP500x2

Table 3-543: Ports

Name	Protocol	Type	Description
dc_de_interrupt[2]	Signal	Master	Interrupt signalling from display engines.
dc_pvbus_m[2]	PVBus	Master	Bus for processor 0 and 1.
dc_se_interrupt[2]	Signal	Master	Interrupt signalling from scaling engines.
display[2]	LCD	Master	Connection to visualization component.
dp0_clk_in	ClockSignal	Slave	Clock signal for DP0.
dp1_clk_in	ClockSignal	Slave	Clock signal for DP1.
pvbus_s	PVBus	Slave	Slave port for register access.
reset_signal	Signal	Slave	Reset signal.

Parameters for DP500x2

`disable_snooping_dma`

Type

bool

Default value

0x0

Disable DMA snooping.

force_frame_rate

Type

int

Default value

0x0

If 0 - the input clock is used as PXL CLOCK, if >0 then the model ensures the screen display is refreshed n times per simulated second.

3.6.4 DP550

ARM DP550 Display Processor. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-544: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for DP550

This model has the following Iris instances:

Table 3-545: DP550 Iris instances

InstanceName	ComponentName
DP550	DP550
DP550.busmaster	PVBusMaster
DP550.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-546: DP550 MTI instances

InstanceName	ComponentName
DP550	DP550
DP550.busmaster	PVBusMaster
DP550.busslave	PVBusSlave

DP550 contains the following CADI targets:

- DP550

About DP550

This component is a model of the DP550 Display Processor. Connect it to a visualization component to view LCD output. This is the single display configuration of DP550. For the dual display configuration, use DP550x2.

The model provides the following functionality:

- All RGB and YUV format parsing.
- Color adjustment in Display Engine (DE).
- Nearest neighbor scaling.
- All layers.
- Alpha blending.
- Memory writeback.
- Inverse gamma adjustment.
- Basic layer (overlay) and register security semantics.

Limitations

- No support for polyphase scaling algorithm. Falls back to nearest neighbor when configured to do so.
- No support for 3D or interlaced video.
- No support for image enhancing functionality.
- No colorspace conversion support.
- No support for two plane YUV memory writeback.

Ports for DP550

Table 3-547: 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.
intr	Signal	Master	Interrupt signal.
intr_se	Signal	Master	Interrupt signal from scaling engine.
pvbuss_m	PVBus	Master	Bus for processor 0.
pvbuss_s	PVBus	Slave	Slave port for register access.
reset_signal	Signal	Slave	Reset signal.

Parameters for DP550

disable_snooping_dma

Type

bool

Default value

0x0

Disable DMA snooping.

dual_mode

Type

bool

Default value

0x0

Whether 2 display mode is supported.

force_frame_rate

Type

int

Default value

0x0

If 0 - the input clock is used as PXLCLK, if >0 then the model ensures the screen display is refreshed n times per simulated second.

set_nprot_nsaaid

Type

int

Default value

0xc00ba98

Configurable NPROT_NSAAID for non-secure NSAAID values.

set_nprot_streamid

Type

int

Default value

0xc00ba98

Configurable NPROT_STREAMID for non-secure STREAMID values.

set_prot_nsaid

Type
int

Default value
0x4003210

Configurable PROT_NSAID for secure NSAID values.

set_prot_streamid

Type
int

Default value
0x4003210

Configurable PROT_STREAMID for secure STREAMID values.

3.6.5 DP550x2

ARM DP550 Display Processor x2. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-548: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for DP550x2

This model has the following Iris instances:

Table 3-549: DP550x2 Iris instances

InstanceName	ComponentName
DP550x2	DP550x2
DP550x2.display_core_0	DP550
DP550x2.display_core_0.busmaster	PVBusMaster
DP550x2.display_core_0.busslave	PVBusSlave
DP550x2.display_core_1	DP550
DP550x2.display_core_1.busmaster	PVBusMaster
DP550x2.display_core_1.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-550: DP550x2 MTI instances

InstanceName	ComponentName
DP550x2.display_core_0	DP550
DP550x2.display_core_0.busmaster	PVBusMaster
DP550x2.display_core_0.busslave	PVBusSlave
DP550x2.display_core_1	DP550
DP550x2.display_core_1.busmaster	PVBusMaster
DP550x2.display_core_1.busslave	PVBusSlave

DP550x2 contains the following CADI targets:

- DP550
- DP550x2

About DP550x2

This component is a model of the dual display configuration of the DP550 Display Processor. Connect it to a visualization component to view LCD output.

The model provides the following functionality:

- All RGB and YUV format parsing.
- Color adjustment in Display Engine (DE).
- Nearest neighbor scaling.
- All layers.
- Alpha blending.
- Memory writeback.
- Inverse gamma adjustment.
- Basic layer (overlay) and register security semantics.

Limitations

- No support for polyphase scaling algorithm. Falls back to nearest neighbor when configured to do so.
- No support for 3D or interlaced video.
- No support for image enhancing functionality.
- No colorspace conversion support.
- No support for two plane YUV memory writeback.

Ports for DP550x2

Table 3-551: Ports

Name	Protocol	Type	Description
dc_de_interrupt[2]	Signal	Master	Interrupt signalling from display engines.

Name	Protocol	Type	Description
dc_pvbus_m[2]	PVBus	Master	Bus for processor 0 and 1.
dc_se_interrupt[2]	Signal	Master	Interrupt signalling from scaling engines.
display[2]	LCD	Master	Connection to visualization component.
dp0_clk_in	ClockSignal	Slave	Clock signal for DP0.
dp1_clk_in	ClockSignal	Slave	Clock signal for DP1.
pvbus_s	PVBus	Slave	Slave port for register access.
reset_signal	Signal	Slave	Reset signal.

Parameters for DP550x2

disable_snooping_dma

Type

bool

Default value

0x0

Disable DMA snooping.

force_frame_rate

Type

int

Default value

0x0

If 0 - the input clock is used as PXLCLK, if >0 then the model ensures the screen display is refreshed n times per simulated second.

set_primary_nprot_nsaidth

Type

int

Default value

0xc00ba98

Configurable NPROT_NSAID for non-secure NSAID values(primary).

set_primary_nprot_streamid

Type

int

Default value

0xc00ba98

Configurable NPROT_STREAMID for non-secure STREAMID values(primary).

set_primary_prot_nsaid**Type**

int

Default value

0x4003210

Configurable PROT_NSAID for secure NSAID values(primary).

set_primary_prot_streamid**Type**

int

Default value

0x4003210

Configurable PROT_STREAMID for secure STREAMID values(primary).

set_secondary_nprot_nsaid**Type**

int

Default value

0xc00ba98

Configurable NPROT_NSAID for non-secure NSAID values(secondary).

set_secondary_nprot_streamid**Type**

int

Default value

0xc00ba98

Configurable NPROT_STREAMID for non-secure STREAMID values(secondary).

set_secondary_prot_nsaid**Type**

int

Default value

0x4003210

Configurable PROT_NSAID for secure NSAID values(secondary).

set_secondary_prot_streamid**Type**

int

Default value

0x4003210

Configurable PROT_STREAMID for secure STREAMID values(secondary).

3.6.6 DP650

ARM DP650 Display Processor. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-552: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for DP650

This model has the following Iris instances:

Table 3-553: DP650 Iris instances

InstanceName	ComponentName
DP650	DP650
DP650.busmaster	PVBusMaster
DP650.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-554: DP650 MTI instances

InstanceName	ComponentName
DP650	DP650
DP650.busmaster	PVBusMaster
DP650.busslave	PVBusSlave

DP650 contains the following CADI targets:

- DP650

About DP650

This component is a model of the DP650 Display Processor. Connect it to a visualization component to view LCD output. This is the single display configuration of DP650. For the dual display configuration, use DP650x2.

Limitations

- No support for the polyphase scaling algorithm, it always uses nearest neighbor.
- No support for 3D or interlaced video.
- No support for image enhancing functionality.
- No YUV 2-plane support for memory writeback.
- No color space conversion support.

Ports for DP650

Table 3-555: 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.
intr	Signal	Master	Interrupt signal.
intr_se	Signal	Master	Interrupt signal from scaling engine.
pvbus_m	PVBus	Master	Bus for processor 0.
pvbus_s	PVBus	Slave	Slave port for register access.
reset_signal	Signal	Slave	Reset signal.

Parameters for DP650

disable_snooping_dma

Type

bool

Default value

0x0

Disable DMA snooping.

dual_mode

Type

bool

Default value

0x0

Whether 2 display mode is supported.

force_frame_rate**Type**

int

Default value

0x0

If 0 - the input clock is used as PXLCLK, if >0 then the model ensures the screen display is refreshed n times per simulated second.

set_nprot_nsaid**Type**

int

Default value

0xc00ba98

Configurable NPROT_NSaid for non-secure NSaid values.

set_nprot_streamid**Type**

int

Default value

0xc00ba98

Configurable NPROT_STREAMID for non-secure STREAMID values.

set_prot_nsaid**Type**

int

Default value

0x4003210

Configurable PROT_NSaid for secure NSaid values.

set_prot_streamid**Type**

int

Default value

0x4003210

Configurable PROT_STREAMID for secure STREAMID values.

3.6.7 DP650x2

ARM DP650 Display Processor x2. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-556: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for DP650x2

This model has the following Iris instances:

Table 3-557: DP650x2 Iris instances

InstanceName	ComponentName
DP650x2	DP650x2
DP650x2.display_core_0	DP650
DP650x2.display_core_0.busmaster	PVBusMaster
DP650x2.display_core_0.busslave	PVBusSlave
DP650x2.display_core_1	DP650
DP650x2.display_core_1.busmaster	PVBusMaster
DP650x2.display_core_1.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-558: DP650x2 MTI instances

InstanceName	ComponentName
DP650x2.display_core_0	DP650
DP650x2.display_core_0.busmaster	PVBusMaster
DP650x2.display_core_0.busslave	PVBusSlave
DP650x2.display_core_1	DP650
DP650x2.display_core_1.busmaster	PVBusMaster
DP650x2.display_core_1.busslave	PVBusSlave

DP650x2 contains the following CADI targets:

- DP650
- DP650x2

About DP650x2

This component is a model of the dual display configuration of the DP650 Display Processor. To view LCD output, connect it to a visualization component.

Limitations

- No support for the polyphase scaling algorithm, it always uses nearest neighbor.
- No support for 3D or interlaced video.
- No support for image enhancing functionality.
- No YUV 2-plane support for memory writeback.
- No color space conversion support.

Ports for DP650x2

Table 3-559: Ports

Name	Protocol	Type	Description
dc_de_interrupt[2]	Signal	Master	Interrupt signalling from display engines.
dc_pvbus_m[2]	PVBus	Master	Bus for processor 0 and 1.
dc_se_interrupt[2]	Signal	Master	Interrupt signalling from scaling engines.
display[2]	LCD	Master	Connection to visualization component.
dp0_clk_in	ClockSignal	Slave	Clock signal for DP0.
dp1_clk_in	ClockSignal	Slave	Clock signal for DP1.
pvbus_s	PVBus	Slave	Slave port for register access.
reset_signal	Signal	Slave	Reset signal.

Parameters for DP650x2

disable_snooping_dma

Type

bool

Default value

0x0

Disable DMA snooping.

force_frame_rate

Type

int

Default value

0x0

If 0 - the input clock is used as PXLCLK, if >0 then the model ensures the screen display is refreshed n times per simulated second.

set_primary_nprot_nsaid

Type

int

Default value

0xc00ba98

Configurable NPROT_NSAID for non-secure NSAID values(primary).

set_primary_nprot_streamid**Type**

int

Default value

0xc00ba98

Configurable NPROT_STREAMID for non-secure STREAMID values(primary).

set_primary_prot_nsaid**Type**

int

Default value

0x4003210

Configurable PROT_NSAID for secure NSAID values(primary).

set_primary_prot_streamid**Type**

int

Default value

0x4003210

Configurable PROT_STREAMID for secure STREAMID values(primary).

set_secondary_nprot_nsaid**Type**

int

Default value

0xc00ba98

Configurable NPROT_NSAID for non-secure NSAID values(secondary).

set_secondary_nprot_streamid**Type**

int

Default value

0xc00ba98

Configurable NPROT_STREAMID for non-secure STREAMID values(secondary).

set_secondary_prot_nsaid

Type
int

Default value
0x4003210

Configurable PROT_NSAID for secure NSAID values(secondary).

set_secondary_prot_streamid

Type
int

Default value
0x4003210

Configurable PROT_STREAMID for secure STREAMID values(secondary).

3.6.8 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 3-560: 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 23.

Iris and MTI instances for Mali_C5x_streaming_sink

This model has the following Iris instances:

Table 3-561: 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 3-562: 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

Mali_C5x_streaming_sink contains the following CADI targets:

- Mali_C5x_streaming_sink

Ports for Mali_C5x_streaming_sink

Table 3-563: 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

Saving captured frames on/off.

fn_prefix

Type

string

Default value

"isp_out_"

Saved filenames prefix.

3.6.9 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 3-564: 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 23.

Iris and MTI instances for Mali_C7x_streaming_sink

This model has the following Iris instances:

Table 3-565: 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 3-566: 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

Mali_C7x_streaming_sink contains the following CADI targets:

- Mali_C7x_streaming_sink

Ports for Mali_C7x_streaming_sink

Table 3-567: 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

Saving captured frames on/off.

fn_prefix

Type

string

Default value

"isp_out_"

Saved filenames prefix.

3.6.10 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 3-568: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Changes in 11.27.17

Parameters added:

- CNR_FITTED
- COMPRESSION_FITTED
- DSPIPE_FITTED
- FRSCALER_FITTED
- IRIDIX_GTM_FITTED
- IRIDIX_LTM_FITTED
- PONG_CONFIG_FITTED
- SCALER_COEF_SETS
- SINTER_FITTED

- SINTER_LITE
- TEMPER_FITTED
- WDR_FITTED
- ext_mode
- verbosity

Parameters removed:

- hw_config

Iris and MTI instances for Mali_C55

This model has the following Iris instances:

Table 3-569: 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 3-570: 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

Mali_C55 contains the following CADI targets:

- Mali_C55

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.

Ports for Mali_C55

Table 3-571: 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
pvbuse_m	PVBus	Master	Master AXI port for RAM access
pvbuse_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

Color Noise Reduction (CNR) and the square and square root for CNR: 0 - absent, 1 - present.

COMPRESSION_FITTED

Type

int

Default value

0x1

Temper compression logic: 0 - absent, 1 - present.

DSPIPE_FITTED

Type

int

Default value

0x1

Downscaled pipeline branch: 0 - absent, 1 - present.

FRSCALER_FITTED**Type**

int

Default value

0x1

Rull Resolution pipeline RGB scaler: 0 - absent, 1 - present.

IRIDIX_GTM_FITTED**Type**

int

Default value

0x0

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

Iridix(TM) local tone-mapping logic: 0 - absent, 1 - present.

PONG_CONFIG_FITTED**Type**

int

Default value

0x1

Pong configuration space: 0 - absent, 1 - present.

SCALER_COEF_SETS**Type**

int

Default value

0x8

Number of scaler coefficient sets (8 or 16).

SINTER_FITTED**Type**

int

Default value

0x1

Sinter block: 0 - absent, 1 - present.

SINTER_LITE**Type**

int

Default value

0x0

Sinter version: 0 - full, 1 - lite.

TEMPER_FITTED**Type**

int

Default value

0x1

Temper, DMA, or merge: 0 - absent, 1 - present.

WDR_FITTED**Type**

int

Default value

0x1

Wide Dynamic Range (WDR) frame stitch, offset, and gain: 0 - absent, 1 - present.

ext_mode**Type**

int

Default value

0x0

Reserved for future use. Use it with instructions from Arm Technical Support (support-esl@arm.com).

verbosity**Type**

int

Default value

0x0

Messages verbosity level: -1 - print ERROR messages only, 0 - print ERROR and WARNING messages, 1 - print INFO messages, 2 - print DEBUG messages.

3.6.11 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 3-572: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Changes in 11.27.17

Parameters added:

- `verbosity`

Iris and MTI instances for Mali_C71

This model has the following Iris instances:

Table 3-573: 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 3-574: Mali_C71 MTI instances

InstanceName	ComponentName
Mali_C71.apb_slave	PVBusSlave
Mali_C71.bus_slave0	PVBusSlave

InstanceName	ComponentName
Mali_C71.bus_slave1	PVBusSlave
Mali_C71.bus_slave2	PVBusSlave
Mali_C71.bus_slave3	PVBusSlave

Mali_C71 contains the following CADI targets:

- Mali_C71

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.

Ports for Mali_C71

Table 3-575: 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_C71

verbosity

Type

int

Default value

0x0

Messages verbosity level: -1 - print ERROR messages only, 0 - print ERROR and WARNING messages, 1 - print INFO messages, 2 - print DEBUG messages.

3.6.12 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 3-576: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Changes in 11.27.17

Parameters added:

- `verbosity`

Iris and MTI instances for Mali_C78

This model has the following Iris instances:

Table 3-577: 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 3-578: 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

Mali_C78 contains the following CADI targets:

- Mali_C78

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.

Ports for Mali_C78

Table 3-579: 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

Messages verbosity level: -1 - print ERROR messages only, 0 - print ERROR and WARNING messages, 1 - print INFO messages, 2 - print DEBUG messages.

3.6.13 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 3-580: 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 23.

Iris and MTI instances for Mali_Cxx_streaming_camera

This model has the following Iris instances:

Table 3-581: 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 3-582: Mali_Cxx_streaming_camera MTI instances

InstanceName	ComponentName
Mali_Cxx_streaming_camera.bus_master_data	PVBusMaster
Mali_Cxx_streaming_camera.bus_slave_config	PVBusSlave

Mali_Cxx_streaming_camera contains the following CADI targets:

- Mali_Cxx_streaming_camera

Ports for Mali_Cxx_streaming_camera

Table 3-583: 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

'''

A file containing one or more frames to stream in the ISP model's FRM format.

3.6.14 Mali_G51

ARM Mali-G51 GPU. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-584: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for Mali_G51

This model has the following Iris instances:

Table 3-585: Mali_G51 Iris instances

InstanceName	ComponentName
Mali_G51	Mali_G51
Mali_G51.busmaster	PVBusMaster
Mali_G51.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-586: Mali_G51 MTI instances

InstanceName	ComponentName
Mali_G51	Mali_G51
Mali_G51.busmaster	PVBusMaster
Mali_G51.busslave	PVBusSlave

Mali_G51 contains the following CADI targets:

- Mali_G51

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.

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.
- 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.

Ports for Mali_G51

Table 3-587: 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.

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 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_G51

revision

Type

string

Default value

"r0p0"

Revision of the RTL that the model represents. Valid values: r0p0.

3.6.15 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 3-588: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for Mali_G71

This model has the following Iris instances:

Table 3-589: 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 3-590: Mali_G71 MTI instances

InstanceName	ComponentName
Mali_G71	Mali_G71

InstanceName	ComponentName
Mali_G71.busmaster	PVBusMaster
Mali_G71.busslave	PVBusSlave

Mali_G71 contains the following CADI targets:

- Mali_G71

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.

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.
- 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.

Ports for Mali_G71

Table 3-591: 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"

Revision of the RTL that the model represents. Valid values: r0p0.

3.6.16 Mali_G72

ARM Mali-G72 GPU. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-592: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for Mali_G72

This model has the following Iris instances:

Table 3-593: Mali_G72 Iris instances

InstanceName	ComponentName
Mali_G72	Mali_G72
Mali_G72.busmaster	PVBusMaster
Mali_G72.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-594: Mali_G72 MTI instances

InstanceName	ComponentName
Mali_G72	Mali_G72
Mali_G72.busmaster	PVBusMaster
Mali_G72.busslave	PVBusSlave

Mali_G72 contains the following CADI targets:

- Mali_G72

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.

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.
- 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.

Ports for Mali_G72

Table 3-595: 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_G72

revision

Type

string

Default value

"r0p0"

Revision of the RTL that the model represents. Valid values: r0p0.

3.6.17 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 3-596: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for Mali_G76

This model has the following Iris instances:

Table 3-597: 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 3-598: Mali_G76 MTI instances

InstanceName	ComponentName
Mali_G76	Mali_G76
Mali_G76.busmaster	PVBusMaster
Mali_G76.busslave	PVBusSlave

Mali_G76 contains the following CADI targets:

- Mali_G76

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.

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 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.

Ports for Mali_G76

Table 3-599: 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.

Name	Protocol	Type	Description
pvbuss_s	PVBus	Slave	The interface for the CPU to access the GPU registers.

Parameters for Mali_G76

revision

Type

string

Default value

"rOp0"

Revision of the RTL that the model represents. Valid values: rOp0.

3.6.18 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 3-600: IP revisions support

Revision	Quality level
rOp0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for Mali_G78AE

This model has the following Iris instances:

Table 3-601: Mali_G78AE Iris instances

InstanceName	ComponentName
Mali_G78AE	Mali_G78AE
Mali_G78AE.AccessControl	PVBusMapper

This model has the following MTI trace components:

Table 3-602: Mali_G78AE MTI instances

InstanceName	ComponentName
Mali_G78AE	Mali_G78AE
Mali_G78AE.AccessControl	PVBusMapper

Mali_G78AE contains the following CADI targets:

- Mali_G78AE

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.

Ports for Mali_G78AE

Table 3-603: 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 irq
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]"

Specification of how the StreamID is encoded into transaction attributes.

3.6.19 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 3-604: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for Mali_G710

This model has the following Iris instances:

Table 3-605: 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 3-606: Mali_G710 MTI instances

InstanceName	ComponentName
Mali_G710	Mali_G710
Mali_G710.busmaster	PVBusMaster
Mali_G710.busslave	PVBusSlave

Mali_G710 contains the following CADI targets:

- Mali_G710

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.

Ports for Mali_G710

Table 3-607: 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

""

Alternate command line for the GPU model. If used, mode is ignored.

altmodel

Type

string

Default value

""

Path to an alternative GPU model library.

mode

Type

string

Default value

"fast"

GPU Mode. Inputs supported: [fast, turbo_fallback, turbo].

3.6.20 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 3-608: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for Mali_G715

This model has the following Iris instances:

Table 3-609: 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 3-610: Mali_G715 MTI instances

InstanceName	ComponentName
Mali_G715	Mali_G715
Mali_G715.busmaster	PVBusMaster
Mali_G715.busslave	PVBusSlave

Mali_G715 contains the following CADI targets:

- Mali_G715

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. See later in this topic for details of supported window systems.
- 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.

Ports for Mali_G715

Table 3-611: 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

""

Alternate command line for the GPU model. If used, mode is ignored.

altmodel

Type

string

Default value

""

Path to an alternative GPU model library.

mode

Type

string

Default value

"fast"

GPU Mode. Inputs supported: [fast, turbo_fallback, turbo].

3.6.21 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 3-612: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for Mali_G720

This model has the following Iris instances:

Table 3-613: 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 3-614: Mali_G720 MTI instances

InstanceName	ComponentName
Mali_G720	Mali_G720
Mali_G720.busmaster	PVBusMaster
Mali_G720.busslave	PVBusSlave

Mali_G720 contains the following CADI targets:

- Mali_G720

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.

Ports for Mali_G720

Table 3-615: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-
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

""

Alternate command line for the GPU model. If used, mode is ignored.

altmodel

Type

string

Default value

""

Path to an alternative GPU model library.

mode

Type

string

Default value

"fast"

GPU Mode. Inputs supported: [fast, turbo_fallback, turbo].

turbo_threads

Type

int

Default value

0x0

Number of threads used.

3.6.22 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 3-616: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for Mali_G725

This model has the following Iris instances:

Table 3-617: 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 3-618: Mali_G725 MTI instances

InstanceName	ComponentName
Mali_G725	Mali_G725
Mali_G725.busmaster	PVBusMaster
Mali_G725.busslave	PVBusSlave

Mali_G725 contains the following CADI targets:

- Mali_G725

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.

Ports for Mali_G725

Table 3-619: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-
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_G725

altcmdline

Type

string

Default value

""

Alternate command line for the GPU model. If used, mode is ignored.

altmodel

Type

string

Default value

""

Path to an alternative GPU model library.

mode

Type

string

Default value

"fast"

GPU Mode. Inputs supported: [fast, turbo_fallback, turbo].

turbo_threads

Type

int

Default value

0x0

Number of threads used.

3.6.23 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 3-620: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for Mali_T624

This model has the following Iris instances:

Table 3-621: 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 3-622: Mali_T624 MTI instances

InstanceName	ComponentName
Mali_T624	Mali_T624
Mali_T624.busmaster	PVBusMaster
Mali_T624.busslave	PVBusSlave

Mali_T624 contains the following CADI targets:

- Mali_T624

Ports for Mali_T624

Table 3-623: 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_T624

revision

Type

string

Default value

"r0p0"

Revision of the RTL that the model represents. Valid values: r0p0.

3.6.24 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 3-624: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for V61

This model has the following Iris instances:

Table 3-625: 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 3-626: V61 MTI instances

InstanceName	ComponentName
V61	V61
V61.BusModifier.LSID0	PVBusMapper
V61.BusModifier.LSID1	PVBusMapper
V61.BusModifier.LSID2	PVBusMapper
V61.BusModifier.LSID3	PVBusMapper

InstanceName	ComponentName
V61.apb_slave[0]	PVBusSlave

V61 contains the following CADI targets:

- V61

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"



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.

Ports for V61

Table 3-627: 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

AXI data width, logarithmic byte notation (3->64bit, 4->128 bit).

enable-frame-rate-limiting

Type

bool

Default value

0x0

enable output rate control via the CLK port.

fuse-disable-AFBC

Type

bool

Default value

0x0

disable AFBC support by fuse.

fuse-disable-HEVC

Type

bool

Default value

0x0

disable HEVC support by fuse.

fuse-disable-Real**Type**

bool

Default value

0x0

disable RealVideo support by fuse.

fuse-disable-VPX**Type**

bool

Default value

0x0

disable VP8 support by fuse.

ncores**Type**

int

Default value

0x1

Number of cores in the component.

omx-library-path**Type**

string

Default value

""

path to a user-provided OMX library; leave blank to use FFomaxIL.

supports-10bit**Type**

bool

Default value

0x1

component supports 10-bit content decoding.

supports-64byte-ref-bursts

Type
bool
Default value
0x1

component supports 64-byte bursts for reference pixel data.

supports-encoding

Type
bool
Default value
0x1

component supports encoding.

3.6.25 V550

Arm® Mali™-V550 Video Processor. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-628: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for V550

This model has the following Iris instances:

Table 3-629: V550 Iris instances

InstanceName	ComponentName
V550	V550
V550.BusModifier.LSID0	PVBusMapper
V550.BusModifier.LSID1	PVBusMapper
V550.BusModifier.LSID2	PVBusMapper
V550.BusModifier.LSID3	PVBusMapper
V550.apb_slave[0]	PVBusSlave

This model has the following MTI trace components:

Table 3-630: V550 MTI instances

InstanceName	ComponentName
V550	V550
V550.BusModifier.LSID0	PVBusMapper
V550.BusModifier.LSID1	PVBusMapper
V550.BusModifier.LSID2	PVBusMapper
V550.BusModifier.LSID3	PVBusMapper
V550.apb_slave[0]	PVBusSlave

V550 contains the following CADI targets:

- V550

About V550

The model requires an external OpenMAX (OMX) IL implementation for codec functionality. By default, V550 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 parameter `omx-library-path`.

FFomaxIL is an OMX IL implementation that Arm provides for convenience in the Third-Party IP (TPIP) add-on package. For more details on FFomaxIL, refer to the TPIP package, which is available from [Product Download Hub](#).

When querying the OMX core, V550 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"



To build example platforms containing V550, 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 and RealVideo decoders.
- No support for 10-bit video output.
- No profiling support.
- No QoS support.
- Power and Test modes are modeled only as register state changes.

Ports for V550

Table 3-631: 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 V550

AXI-data-width

Type

int

Default value

0x4

AXI data width, logarithmic byte notation (3->64bit, 4->128 bit).

enable-frame-rate-limiting

Type

bool

Default value

0x0

enable output rate control via the CLK port.

fuse-disable-AFBC**Type**

bool

Default value

0x0

disable AFBC support by fuse.

fuse-disable-Real**Type**

bool

Default value

0x0

disable RealVideo support by fuse.

fuse-disable-VP8**Type**

bool

Default value

0x0

disable VP8 support by fuse.

ncores**Type**

int

Default value

0x1

Number of cores in the component.

omx-library-path**Type**

string

Default value

""

path to a user-provided OMX library; leave blank to use FFomaxIL.

supports-10bit

Type
bool

Default value
0x1

component supports 10-bit content decoding.

supports-VP8

Type
bool

Default value
0x1

component supports VP8.

supports-encoding

Type
bool

Default value
0x1

component supports encoding.

3.7 Peripheral components

This section describes the Peripheral components.

3.7.1 AudioOut_File

File based Audio Output for PL041_AACI. This model is written in LISA+.

Iris and MTI instances for AudioOut_File

This model has the following Iris instances:

Table 3-632: AudioOut_File Iris instances

InstanceName	ComponentName
AudioOut_File	AudioOut_File

AudioOut_File contains the following CADI targets:

- AudioOut_File

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.

Ports for AudioOut_File

Table 3-633: 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
 ""

Filename.

3.7.2 AudioOut_SDL

SDL based Audio Output for PL041_AACI. This model is written in LISA+.

Iris and MTI instances for AudioOut_SDL

This model has the following Iris instances:

Table 3-634: AudioOut_SDL Iris instances

InstanceName	ComponentName
AudioOut_SDL	AudioOut_SDL

AudioOut_SDL contains the following CADI targets:

- AudioOut_SDL

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.

Ports for AudioOut_SDL

Table 3-635: Ports

Name	Protocol	Type	Description
audio	AudioControl	Slave	Audio input for a connection to a component such as the PL041_AACI.

3.7.3 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 3-636: 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
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.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 3-637: 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

Base_PowerController contains the following CADI targets:

- Base_PowerController
- ClockTimerThread
- ClockTimerThread64
- SchedulerThread
- SchedulerThreadEvent

Ports for Base_PowerController

Table 3-638: Ports

Name	Protocol	Type	Description
cpuporeset[36]	Signal	Master	-
dbgnopwrdown[36]	Signal	Slave	-
delayed_ppu_transition_handler	TimerCallback	Slave	-
l2reset[4]	Signal	Master	-
pchannel_m[36]	PChannel	Master	-
pvbus_s	PVBus	Slave	-
standbywfi[36]	Signal	Slave	-
standbywfi12[4]	Signal	Slave	-
system_reset	Signal	Master	-
system_reset_req	Signal	Slave	-
timer_callback_handler	TimerCallback	Slave	Handle event notifications from the timer.
utility_bus_m[4]	PVBus	Master	-
wakerequest[36]	Signal	Slave	-

Parameters for Base_PowerController

Affinity-shifted

Type

bool

Default value

0x0

Whether core number is reflected in Affinity1 instead of Affinity0.

CPU-affinities

Type

string

Default value

"0.0.0.0"

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

0xffffffffffffffff

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

If lock step is enabled, the number of available cores get reduced to half.

startup**Type**

string

Default value

"0.0.0.*"

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

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

Set this to true if the pchannel is connected to cpus with thread support.

3.7.4 CMSDK_Watchdog_cpp

ARM Watchdog Module. This model is written in C++.

Iris and MTI instances for CMSDK_Watchdog_cpp

This model has the following Iris instances:

Table 3-639: CMSDK_Watchdog_cpp Iris instances

InstanceName	ComponentName
CMSDK_Watchdog_cpp	CMSDK_Watchdog
CMSDK_Watchdog_cpp.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-640: CMSDK_Watchdog_cpp MTI instances

InstanceName	ComponentName
CMSDK_Watchdog_cpp.PVBusSlave	PVBusSlave

CMSDK_Watchdog_cpp contains the following CADI targets:

- CMSDK_Watchdog

Ports for CMSDK_Watchdog_cpp

Table 3-641: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-
irq_out	Signal	Master	-
pvbus_s	PVBus	Slave	-
reset_in	Signal	Slave	-
reset_out	Signal	Master	-

Parameters for CMSDK_Watchdog_cpp

diagnostics

Type

int

Default value

0x2

Diagnostics.

simhalt

Type

bool

Default value

0x0

Halt on reset.

3.7.5 Cluster_Temperature_Sensor

Component to calculate the temperature value of all cores in a cluster. This model is written in C++.

Changes in 11.27.17

Parameters added:

- tempCoeff_A
- tempCoeff_B
- tempCoeff_K
- tempCoeff_TAMB

Parameters removed:

- A
- B
- K
- TAMB

Iris and MTI instances for Cluster_Temperature_Sensor

This model has the following Iris instances:

Table 3-642: Cluster_Temperature_Sensor Iris instances

InstanceName	ComponentName
Cluster_Temperature_Sensor	Cluster_Temperature_Sensor

Cluster_Temperature_Sensor contains the following CADI targets:

- Cluster_Temperature_Sensor

Ports for Cluster_Temperature_Sensor

Table 3-643: 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	-

Name	Protocol	Type	Description
temperature_out	ValueState	Master	-

Parameters for Cluster_Temperature_Sensor

MAX_FREQ

Type

int

Default value

0x77359400

Maximum frequency (in Hz) at which each core can run.

NUM_CORES

Type

int

Default value

0x4

Number of cores per cluster.

diagnostics

Type

int

Default value

0x0

Diagnostics.

tempCoeff_A

Type

string

Default value

"0.5"

Temperature Coefficient.

tempCoeff_B

Type

string

Default value

"0.5"

Temperature Coefficient.

tempCoeff_K

Type

int

Default value

0x32

Temperature Coefficient.

tempCoeff_TAMB

Type

int

Default value

0x14

Temperature Coefficient.

3.7.6 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 3-644: CombinedMessagingUnit Iris instances

InstanceName	ComponentName
CombinedMessagingUnit	CombinedMessagingUnit
CombinedMessagingUnit.host_to_local	MessageHandlingUnit
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

InstanceName	ComponentName
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 3-645: 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
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

CombinedMessagingUnit contains the following CADI targets:

- CombinedMessagingUnit
- MessageHandlingUnit
- MessageHandlingUnitV2

- MessageHandlingUnitV3
- PVBusLogger
- SignalLogger

Ports for CombinedMessagingUnit

Table 3-646: 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

Number of doorbell channels.

NUM_FAST_CH

Type

int

Default value

0x1

Number of fast channels.

diagnostics

Type

int

Default value

0x2

Diagnostics 0==FATAL_ERROR -> 4==DEBUG.

fast_ch_group_int_enable**Type**

bool

Default value

0x0

Fast Channel group interrupts enable, default=false.

fast_ch_n_per_group**Type**

int

Default value

0x1

Fast Channel num channels per group, default=1.

fast_ch_num_groups**Type**

int

Default value

0x1

Fast Channel num of groups, default=1.

fast_ch_word_size**Type**

int

Default value

0x20

Fast Channel word size 32bit or 64bit, default=32.

host_to_local.a_to_b_v3.NUM_FIFO_CH**Type**

int

Default value

0x1

Number of FIFO Channels, default=1.

host_to_local.a_to_b_v3.auto_op_full**Type**

bool

Default value

0x0

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

Fast Channel group interrupts enable, default=false.

host_to_local.a_to_b_v3.fifo_depth**Type**

int

Default value

0x4

Depth of the FIFO = fifo_depth + 1.

host_to_local.a_to_b_v3.m16ba_spt**Type**

bool

Default value

0x0

Mailbox 16 bit access support to FIFO registers.

host_to_local.a_to_b_v3.m32ba_spt**Type**

bool

Default value

0x1

Mailbox 32 bit access support to FIFO registers.

host_to_local.a_to_b_v3.m64ba_spt**Type**

bool

Default value

0x0

Mailbox 64 bit access support to FIFO registers.

host_to_local.a_to_b_v3.m8ba_spt**Type**

bool

Default value

0x0

Mailbox 8 bit access support to FIFO registers.

host_to_local.a_to_b_v3.monolithic**Type**

bool

Default value

0x1

Monolithic or Distributed MHU - default: monolithic(true).

host_to_local.a_to_b_v3.p16ba_spt**Type**

bool

Default value

0x0

Postbox 16 bit access support to FIFO registers.

host_to_local.a_to_b_v3.p32ba_spt**Type**

bool

Default value

0x1

Postbox 32 bit access support to FIFO registers.

host_to_local.a_to_b_v3.p64ba_spt**Type**

bool

Default value

0x0

Postbox 64 bit access support to FIFO registers.

host_to_local.a_to_b_v3.p8ba_spt**Type**

bool

Default value

0x0

Postbox 8 bit access support to FIFO registers.

host_to_local.fast_ch_group_int_enable**Type**

bool

Default value

0x0

Fast Channel group interrupts enable, default=false.

host_to_local_rcv_log.trace_snoops**Type**

bool

Default value

0x0

Enable tracing of ACE snoop requests.

host_to_local_snd_log.trace_snoops**Type**

bool

Default value

0x0

Enable tracing of ACE snoop requests.

irq_rcv_combined_host_log.forward_signal**Type**

bool

Default value

0x1

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

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

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

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

Number of FIFO Channels, default=1.

local_to_host.a_to_b_v3.auto_op_full**Type**

bool

Default value

0x0

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

Fast Channel group interrupts enable, default=false.

local_to_host.a_to_b_v3.fifo_depth**Type**

int

Default value

0x4

Depth of the FIFO = fifo_depth + 1.

local_to_host.a_to_b_v3.m16ba_spt**Type**

bool

Default value

0x0

Mailbox 16 bit access support to FIFO registers.

local_to_host.a_to_b_v3.m32ba_spt**Type**

bool

Default value

0x1

Mailbox 32 bit access support to FIFO registers.

local_to_host.a_to_b_v3.m64ba_spt**Type**

bool

Default value

0x0

Mailbox 64 bit access support to FIFO registers.

local_to_host.a_to_b_v3.m8ba_spt**Type**

bool

Default value

0x0

Mailbox 8 bit access support to FIFO registers.

local_to_host.a_to_b_v3.monolithic**Type**

bool

Default value

0x1

Monolithic or Distributed MHU - default: monolithic(true).

local_to_host.a_to_b_v3.p16ba_spt**Type**

bool

Default value

0x0

Postbox 16 bit access support to FIFO registers.

local_to_host.a_to_b_v3.p32ba_spt**Type**

bool

Default value

0x1

Postbox 32 bit access support to FIFO registers.

local_to_host.a_to_b_v3.p64ba_spt**Type**

bool

Default value

0x0

Postbox 64 bit access support to FIFO registers.

local_to_host.a_to_b_v3.p8ba_spt**Type**

bool

Default value

0x0

Postbox 8 bit access support to FIFO registers.

local_to_host.fast_ch_group_int_enable**Type**

bool

Default value

0x0

Fast Channel group interrupts enable, default=false.

local_to_host_rcv_log.trace_snoops**Type**

bool

Default value

0x0

Enable tracing of ACE snoop requests.

local_to_host_snd_log.trace_snoops**Type**

bool

Default value

0x0

Enable tracing of ACE snoop requests.

major_version

Type
int

Default value
0x2

MHU major version (default=2).

mhu_arch_beta01

Type
bool

Default value
0x0

true = Aligns to MHUv3.beta01; false = Aligns to MHUv3.2.

minor_version

Type
int

Default value
0x1

MHU minor version (default=1).

product_id

Type
int

Default value
0x0

MHU part number.

3.7.7 DebugAccessPort

This model is written in C++.

Ports for DebugAccessPort

Table 3-647: Ports

Name	Protocol	Type	Description
ap_pvbus_m[2]	PVBus	Master	-
clock	ClockSignal	Slave	-

Name	Protocol	Type	Description
paddrdbg31	Signal	Master	-

Parameters for DebugAccessPort

ap0_has_debug_rom

Type

bool

Default value

0x0

Whether AP0 has a Debug ROM.

ap0_rom_base_address

Type

int

Default value

0x0

ROM base address for AP 0.

ap0_set_paddrdbg31

Type

bool

Default value

0x0

Set paddrdbg31 signal during accesses on AP0.

ap1_has_debug_rom

Type

bool

Default value

0x0

Whether AP1 has a Debug ROM.

ap1_rom_base_address

Type

int

Default value

0x0

ROM base address for AP 1.

ap1_set_paddrdbg31

Type
bool

Default value
0x0

Set paddrdbg31 signal during accesses on AP1.

3.7.8 DebugROM

This model is written in C++.

Ports for DebugROM

Table 3-648: Ports

Name	Protocol	Type	Description
paddrdbg31	Signal	Master	-
pvbus_s	PVBus	Slave	-

Parameters for DebugROM

customer_modified

Type
int

Default value
0x0

.

entry_0

Type
int

Default value
0x0

Offset of component 0.

entry_1

Type
int

Default value

0x0

Offset of component 1.

entry_10**Type**

int

Default value

0x0

Offset of component 10.

entry_11**Type**

int

Default value

0x0

Offset of component 11.

entry_12**Type**

int

Default value

0x0

Offset of component 12.

entry_13**Type**

int

Default value

0x0

Offset of component 13.

entry_14**Type**

int

Default value

0x0

Offset of component 14.

entry_15**Type**

int

Default value

0x0

Offset of component 15.

entry_16**Type**

int

Default value

0x0

Offset of component 16.

entry_17**Type**

int

Default value

0x0

Offset of component 17.

entry_18**Type**

int

Default value

0x0

Offset of component 18.

entry_19**Type**

int

Default value

0x0

Offset of component 19.

entry_2**Type**

int

Default value

0x0

Offset of component 2.

entry_20**Type**

int

Default value

0x0

Offset of component 20.

entry_21**Type**

int

Default value

0x0

Offset of component 21.

entry_22**Type**

int

Default value

0x0

Offset of component 22.

entry_23**Type**

int

Default value

0x0

Offset of component 23.

entry_24**Type**

int

Default value

0x0

Offset of component 24.

entry_25**Type**

int

Default value

0x0

Offset of component 25.

entry_26**Type**

int

Default value

0x0

Offset of component 26.

entry_27**Type**

int

Default value

0x0

Offset of component 27.

entry_28**Type**

int

Default value

0x0

Offset of component 28.

entry_29**Type**

int

Default value

0x0

Offset of component 29.

entry_3**Type**

int

Default value

0x0

Offset of component 3.

entry_30**Type**

int

Default value

0x0

Offset of component 30.

entry_31**Type**

int

Default value

0x0

Offset of component 31.

entry_32**Type**

int

Default value

0x0

Offset of component 32.

entry_33**Type**

int

Default value

0x0

Offset of component 33.

entry_34**Type**

int

Default value

0x0

Offset of component 34.

entry_35**Type**

int

Default value

0x0

Offset of component 35.

entry_36**Type**

int

Default value

0x0

Offset of component 36.

entry_37**Type**

int

Default value

0x0

Offset of component 37.

entry_38**Type**

int

Default value

0x0

Offset of component 38.

entry_39**Type**

int

Default value

0x0

Offset of component 39.

entry_4**Type**

int

Default value

0x0

Offset of component 4.

entry_40**Type**

int

Default value

0x0

Offset of component 40.

entry_41**Type**

int

Default value

0x0

Offset of component 41.

entry_42**Type**

int

Default value

0x0

Offset of component 42.

entry_43**Type**

int

Default value

0x0

Offset of component 43.

entry_44**Type**

int

Default value

0x0

Offset of component 44.

entry_45**Type**

int

Default value

0x0

Offset of component 45.

entry_46**Type**

int

Default value

0x0

Offset of component 46.

entry_47**Type**

int

Default value

0x0

Offset of component 47.

entry_48**Type**

int

Default value

0x0

Offset of component 48.

entry_49**Type**

int

Default value

0x0

Offset of component 49.

entry_5**Type**

int

Default value

0x0

Offset of component 5.

entry_50**Type**

int

Default value

0x0

Offset of component 50.

entry_51**Type**

int

Default value

0x0

Offset of component 51.

entry_52**Type**

int

Default value

0x0

Offset of component 52.

entry_53**Type**

int

Default value

0x0

Offset of component 53.

entry_54**Type**

int

Default value

0x0

Offset of component 54.

entry_55**Type**

int

Default value

0x0

Offset of component 55.

entry_56**Type**

int

Default value

0x0

Offset of component 56.

entry_57**Type**

int

Default value

0x0

Offset of component 57.

entry_58**Type**

int

Default value

0x0

Offset of component 58.

entry_59**Type**

int

Default value

0x0

Offset of component 59.

entry_6**Type**

int

Default value

0x0

Offset of component 6.

entry_60**Type**

int

Default value

0x0

Offset of component 60.

entry_61**Type**

int

Default value

0x0

Offset of component 61.

entry_62**Type**

int

Default value

0x0

Offset of component 62.

entry_63**Type**

int

Default value

0x0

Offset of component 63.

entry_7**Type**

int

Default value

0x0

Offset of component 7.

entry_8**Type**

int

Default value

0x0

Offset of component 8.

entry_9**Type**

int

Default value

0x0

Offset of component 9.

manufacturer_revision_number**Type**

int

Default value

0x0

.

part_number**Type**

int

Default value

0x0

.

revision**Type**

int

Default value

0x0

.

3.7.9 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 3-649: DualClusterSystemConfigurationBlock Iris instances

InstanceName	ComponentName
DualClusterSystemConfigurationBlock	DualClusterSystemConfigurationBlock
DualClusterSystemConfigurationBlock.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Table 3-650: DualClusterSystemConfigurationBlock MTI instances

InstanceName	ComponentName
DualClusterSystemConfigurationBlock	DualClusterSystemConfigurationBlock
DualClusterSystemConfigurationBlock.pvbusslave	PVBusSlave

DualClusterSystemConfigurationBlock contains the following CADI targets:

- DualClusterSystemConfigurationBlock

Ports for DualClusterSystemConfigurationBlock

Table 3-651: 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	-
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	-

Name	Protocol	Type	Description
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

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

ClusterId for primary cluster (Cortex-A15) on power-on reset.

Cluster1IdOnPOReset

Type

int

Default value

0x1

ClusterId for secondary cluster (Cortex-A7) on power-on reset.

DCSCB_PERIPHBASE**Type**

int

Default value

0x1e000000

PERIPHBASE.

DCS_AID**Type**

int

Default value

0x0

DCS_AID is the Auxiliary ID Register.

DCS_ID**Type**

int

Default value

0x41120000

The value returned by the DCS_ID register.

DCS_ID_BUILD_NUMBER**Type**

int

Default value

0x1

DCS_ID build number.

DCS_LEDS**Type**

int

Default value

0x0

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

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

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

Number of custom IRQs controlled by interrupt generator is $\text{INTGEN_INTS} * 32 + 32$.

NumberOfCoresInCluster0**Type**

int

Default value

0x2

The number of cores in the primary cluster.

NumberOfCoresInCluster1**Type**

int

Default value

0x2

The number of cores in the secondary cluster.

ResetValueOfDaughterUserSwitches

Type
int
Default value
0x0

Reset value of the user switches on the daughterboard.

stop_on_sequence_id

Type
int
Default value
0x0

If non-zero the sequence_id of the SW trace mechanism on which to halt the simulator.

3.7.10 DummyAPB

DummyAPB. This model is written in LISA+.

Iris and MTI instances for DummyAPB

This model has the following Iris instances:

Table 3-652: DummyAPB Iris instances

InstanceName	ComponentName
DummyAPB	DummyAPB
DummyAPB.pvbuslave	PVBusSlave

This model has the following MTI trace components:

Table 3-653: DummyAPB MTI instances

InstanceName	ComponentName
DummyAPB.pvbuslave	PVBusSlave

DummyAPB contains the following CADI targets:

- DummyAPB

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 need also to set `periph_framesize` to 4 or 64, depending on whether the peripheral has its registers in a 4KB or 64KB 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.

Ports for DummyAPB

Table 3-654: Ports

Name	Protocol	Type	Description
pvbuss_s	PVBus	Slave	Bus slave interface.

Parameters for DummyAPB

fail
Type
bool
Default value
0x0

Abort all accesses.

failmsg
Type
string
Default value
""

String to print when 'fail'=true and access occurred.

periph_framesize**Type**

int

Default value

0xffffffffffffffffffff

Size of frame (4/64, indicating if ID is at xFD0 or xFFD0).

periphid_24**Type**

string

Default value

""

24 hex digits for the 12 bytes of peripheral ID.

periphid_generic**Type**

string

Default value

""

Set of space-separated offset:value pairs for dwords of ID.

ram_generic**Type**

string

Default value

""

Set of space-separated offset:default pairs for writable dwords.

warn_once**Type**

bool

Default value

0x1

Warn once for the invalid read and write access.

3.7.11 ElfLoader

ELF loader component. This model is written in LISA+.

Iris and MTI instances for ElfLoader

This model has the following Iris instances:

Table 3-655: ElfLoader Iris instances

InstanceName	ComponentName
ElfLoader	ElfLoader
ElfLoader.pvbus_busmaster	PVBusMaster

This model has the following MTI trace components:

Table 3-656: ElfLoader MTI instances

InstanceName	ComponentName
ElfLoader.pvbus_busmaster	PVBusMaster

ElfLoader contains the following CADI targets:

- ElfLoader

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

Ports for ElfLoader

Table 3-657: 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

""

ELF file.

impdef_copy**Type**

bool

Default value

0x0

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

""

load file for large address mapping.

ns_copy**Type**

bool

Default value

0x1

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"

Encoding of various attributes on the bus.

realm_copy**Type**

bool

Default value

0x0

load ELF file to REALM memory spaces, if load file is not specified.

root_copy

Type
bool

Default value
0x0

load ELF file to ROOT memory spaces, if load file is not specified.

3.7.12 FlashLoader

A device that can preload a gzipped image into flash at startup. This model is written in C++.

Iris and MTI instances for FlashLoader

This model has the following Iris instances:

Table 3-658: FlashLoader Iris instances

InstanceName	ComponentName
FlashLoader	FlashLoader

FlashLoader contains the following CADI targets:

- FlashLoader

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.

Ports for FlashLoader

Table 3-659: 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

Diagnostics.

fname
Type
string
Default value
"(none)"

Filename (Default '(none)' means: Do not load any file. An empty string will cause a warning.).

fnameWrite
Type
string
Default value
"(none)"

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

write_flash_after_reset.

3.7.13 GICv3CommsLogger

Traces GICv3Comms activity. This model is written in LISA+.

Iris and MTI instances for GICv3CommsLogger

This model has the following Iris instances:

Table 3-660: GICv3CommsLogger Iris instances

InstanceName	ComponentName
GICv3CommsLogger	GICv3CommsLogger

This model has the following MTI trace components:

Table 3-661: GICv3CommsLogger MTI instances

InstanceName	ComponentName
GICv3CommsLogger	GICv3CommsLogger

GICv3CommsLogger contains the following CADI targets:

- GICv3CommsLogger

Ports for GICv3CommsLogger

Table 3-662: 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

Print tracing information to attached debugger in addition to via MTI.

3.7.14 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 3-663: GICv3CommsPVBus Iris instances

InstanceName	ComponentName
GICv3CommsPVBus	GICv3CommsPVBus
GICv3CommsPVBus.bus_slave	PVBusSlave

This model has the following MTI trace components:

Table 3-664: GICv3CommsPVBus MTI instances

InstanceName	ComponentName
GICv3CommsPVBus	GICv3CommsPVBus
GICv3CommsPVBus.bus_slave	PVBusSlave

GICv3CommsPVBus contains the following CADI targets:

- GICv3CommsPVBus

Ports for GICv3CommsPVBUS

Table 3-665: 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	-

3.7.15 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 3-666: GICv3ProtocolChecker Iris instances

InstanceName	ComponentName
GICv3ProtocolChecker	GICv3ProtocolChecker

This model has the following MTI trace components:

Table 3-667: GICv3ProtocolChecker MTI instances

InstanceName	ComponentName
GICv3ProtocolChecker	GICv3ProtocolChecker

GICv3ProtocolChecker contains the following CADI targets:

- GICv3ProtocolChecker

Ports for GICv3ProtocolChecker

Table 3-668: 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

Cpu interface id to which this component is connected.

enable_protocol_checking

Type
bool
Default value
0x1

Enable/disable the protocol checking.

3.7.16 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 `callback` method of the callback port is invoked periodically, at approximately the rate determined by the `delay_ms` parameter.



This is a real-time callback, 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 3-669: 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

GUI update period in ms.

has_gui

Type
bool

Default value
0x1

GUI is enabled.

3.7.17 HostBridge

Host Socket Interface Component. This model is written in C++.

Iris and MTI instances for HostBridge

This model has the following Iris instances:

Table 3-670: HostBridge Iris instances

InstanceName	ComponentName
HostBridge	HostBridge

HostBridge contains the following CADI targets:

- HostBridge

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 Microsoft Windows](#)
- [Configuring the networking environment for Linux](#)
- [User mode networking](#)

Ports for HostBridge

Table 3-671: Ports

Name	Protocol	Type	Description
eth	VirtualEthernet	Slave	-
state	ValueState_64	Slave	-

Parameters for HostBridge

interfaceName

Type

string

Default value

""

Host Interface.

userNetOptions

Type

string

Default value

""

Control options for UserNet TCP/IP (for internal use only, please do not use).

userNetPorts

Type

string

Default value

""

Listening ports to expose in user-mode networking.

userNetSubnet

Type

string

Default value

"172.20.51.0/24"

Virtual subnet for user-mode networking.

userNetworking

Type

bool

Default value

0x0

Enable user-mode networking.

3.7.18 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 3-672: HostSerialInterface Iris instances

InstanceName	ComponentName
HostSerialInterface	HostSerialInterface

HostSerialInterface contains the following CADI targets:

- HostSerialInterface

Ports for HostSerialInterface

Table 3-673: Ports

Name	Protocol	Type	Description
SerialData	SerialData	Slave	Serial data connection to export to host machine.

Parameters for HostSerialInterface

baud_rate

Type

int

Default value

0x0

Baud rate override.

device**Type**

string

Default value

"/dev/ttyS0"

HW device to use.

3.7.19 Infra6_Core_Mgr_Clock_Ctrl

InfraSysDesgin6.x Core Manager and Clock Control register block. This model is written in C++.

Iris and MTI instances for Infra6_Core_Mgr_Clock_Ctrl

This model has the following Iris instances:

Table 3-674: Infra6_Core_Mgr_Clock_Ctrl Iris instances

InstanceName	ComponentName
Infra6_Core_Mgr_Clock_Ctrl	Infra6_core_manager_clock_control_registers
Infra6_Core_Mgr_Clock_Ctrl.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-675: Infra6_Core_Mgr_Clock_Ctrl MTI instances

InstanceName	ComponentName
Infra6_Core_Mgr_Clock_Ctrl.PVBusSlave	PVBusSlave

Infra6_Core_Mgr_Clock_Ctrl contains the following CADI targets:

- [Infra6_core_manager_clock_control_registers](#)

Ports for Infra6_Core_Mgr_Clock_Ctrl

Table 3-676: Ports

Name	Protocol	Type	Description
ap_aarch64_rvbar	Value_64	Master	-
ap_cfgend	Signal	Master	-
ap_l0gptsz	Value	Master	-
core_clkdiv	ClockRateControl	Master	-
core_clken	Signal	Master	-
core_clkssel	Value	Master	-
pvbus_s	PVBus	Slave	-
reset_in	Signal	Slave	-

Parameters for `Infra6_Core_Mgr_Clock_Ctrl`

`cap_clussync`

Type

bool

Default value

0x1

CAP.CLUSSYNC config.

`cap_sync`

Type

int

Default value

0x0

CAP[0].SYNC config.

`diagnostics`

Type

int

Default value

0x0

Diagnostics.

3.7.20 `Infra6_DVFS_Manager`

LCP DVFS Manager. This model is written in C++.

Iris and MTI instances for `Infra6_DVFS_Manager`

This model has the following Iris instances:

Table 3-677: `Infra6_DVFS_Manager` Iris instances

InstanceName	ComponentName
<code>Infra6_DVFS_Manager</code>	<code>Infra6_DVFS_Manager</code>
<code>Infra6_DVFS_Manager.PVBusSlave</code>	<code>PVBusSlave</code>

This model has the following MTI trace components:

Table 3-678: Infra6_DVFS_Manager MTI instances

InstanceName	ComponentName
Infra6_DVFS_Manager.PVBusSlave	PVBusSlave

Infra6_DVFS_Manager contains the following CADI targets:

- Infra6_DVFS_Manager

Ports for Infra6_DVFS_Manager

Table 3-679: Ports

Name	Protocol	Type	Description
core_clkctrl	ClockRateControl	Master	-
pdbus_s	PVBus	Slave	-

Parameters for Infra6_DVFS_Manager

diagnostics

Type

int

Default value

0x2

Diagnostics.

unmapped_range.abort_on_reads

Type

bool

Default value

0x0

Generate Abort on reads.

unmapped_range.abort_on_writes

Type

bool

Default value

0x0

Generate Abort on writes.

unmapped_range.read_data

Type

int

Default value

0x0

Data to return on reads, if not aborting.

unmapped_range.warn_on_reads

Type

bool

Default value

0x1

Generate Warn on reads.

unmapped_range.warn_on_writes

Type

bool

Default value

0x1

Generate Warn on writes.

unmapped_range.warning

Type

string

Default value

"Invalid access"

Warning string.

3.7.21 Infra6_LCP_Extended_Control

LCP Subsystem Control Register Block. This model is written in C++.

Iris and MTI instances for Infra6_LCP_Extended_Control

This model has the following Iris instances:

Table 3-680: Infra6_LCP_Extended_Control Iris instances

InstanceName	ComponentName
Infra6_LCP_Extended_Control	Infra6_LCP_Extended_Control
Infra6_LCP_Extended_Control.PVBusMapper	PVBusMapper
Infra6_LCP_Extended_Control.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-681: Infra6_LCP_Extended_Control MTI instances

InstanceName	ComponentName
Infra6_LCP_Extended_Control.PVBusMapper	PVBusMapper
Infra6_LCP_Extended_Control.PVBusSlave	PVBusSlave

Infra6_LCP_Extended_Control contains the following CADI targets:

- Infra6_LCP_Extended_Control

Ports for Infra6_LCP_Extended_Control

Table 3-682: Ports

Name	Protocol	Type	Description
ap_cluster_ppu_irq_in	Signal	Slave	-
ap_cluster_ppu_irq_out_lcp	Signal	Master	-
ap_cluster_ppu_irq_out_scp	Signal	Master	-
ap_core_ppu_irq_in	Signal	Slave	-
ap_core_ppu_irq_out_lcp	Signal	Master	-
ap_core_ppu_irq_out_scp	Signal	Master	-
pvbus_s	PVBus	Slave	-
tcm_acc_pvbus_m	PVBus	Master	-
tcm_acc_pvbus_s	PVBus	Slave	-
warm_reset_req	Signal	Slave	-

Parameters for Infra6_LCP_Extended_Control

LCP_ID

Type

int

Default value

0x0

LCP Identifier.

diagnostics

Type

int

Default value

0x2

Diagnostics.

3.7.22 Infra6_LCP_External_Control

LCP Control Registers Block. This model is written in C++.

Iris and MTI instances for Infra6_LCP_External_Control

This model has the following Iris instances:

Table 3-683: Infra6_LCP_External_Control Iris instances

InstanceName	ComponentName
Infra6_LCP_External_Control	Infra6_LCP_External_Control
Infra6_LCP_External_Control.PVBusMapper	PVBusMapper
Infra6_LCP_External_Control.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-684: Infra6_LCP_External_Control MTI instances

InstanceName	ComponentName
Infra6_LCP_External_Control.PVBusMapper	PVBusMapper
Infra6_LCP_External_Control.PVBusSlave	PVBusSlave

Infra6_LCP_External_Control contains the following CADI targets:

- [Infra6_LCP_External_Control](#)

Ports for Infra6_LCP_External_Control

Table 3-685: Ports

Name	Protocol	Type	Description
access_status_irq_out	Signal	Master	-
ap2lcp_mhu_irq_in	Signal	Slave	-
ap2lcp_mhu_irq_out_lcp	Signal	Master	-
ap2lcp_mhu_irq_out_scp	Signal	Master	-
comb_irq_stat_in[32]	Signal	Slave	-
combined_irq_out	Signal	Master	-
lockup	Signal	Slave	-
pvbus_s	PVBus	Slave	-
reset_req	Signal	Master	-
uart_filter_pvbus_m	PVBus	Master	-
uart_filter_pvbus_s	PVBus	Slave	-
uart_irq_in	Signal	Slave	-
uart_irq_out	Signal	Master	-

Parameters for Infra6_LCP_External_Control

diagnostics

Type

int

Default value

0x2

Diagnostics.

3.7.23 Infra6_MSCP_PIK

InfraSysDesign6.x MSCP Power Integration Kit. This model is written in C++.

Iris and MTI instances for Infra6_MSCP_PIK

This model has the following Iris instances:

Table 3-686: Infra6_MSCP_PIK Iris instances

InstanceName	ComponentName
Infra6_MSCP_PIK	Infra6_MSCP_PIK
Infra6_MSCP_PIK.Kits2_Timer_cpp	Kits2_Timer
Infra6_MSCP_PIK.Kits2_Timer_cpp.ClockDivider	ClockDivider
Infra6_MSCP_PIK.Kits2_Timer_cpp.CounterModule_cpp	CounterModule
Infra6_MSCP_PIK.PVBusModifier	PVBusMapper
Infra6_MSCP_PIK.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-687: Infra6_MSCP_PIK MTI instances

InstanceName	ComponentName
Infra6_MSCP_PIK.Kits2_Timer_cpp.ClockDivider	ClockDivider
Infra6_MSCP_PIK.PVBusModifier	PVBusMapper
Infra6_MSCP_PIK.PVBusSlave	PVBusSlave

Infra6_MSCP_PIK contains the following CADI targets:

- Infra6_MSCP_PIK

Ports for Infra6_MSCP_PIK

Table 3-688: Ports

Name	Protocol	Type	Description
axi_clkdiv	ClockRateControl	Master	-
axi_clken	Signal	Master	-

Name	Protocol	Type	Description
axi_clkssel	Value	Master	-
clkdiv	ClockRateControl	Master	-
clken	Signal	Master	-
clkssel	Value	Master	-
clock_in	ClockSignal	Slave	-
clus_ppu_int_status[16]	Signal	Slave	-
core_clkdiv	ClockRateControl	Master	-
core_clken	Signal	Master	-
core_clkssel	Value	Master	-
cpu_pll_lock_status[16]	Signal	Slave	-
cpu_pll_unlock_status[16]	Signal	Slave	-
cpu_ppu_int_status[16]	Signal	Slave	-
cpu_warm_reset_out	Signal	Master	-
fhi_irq_out	Signal	Master	-
gtsy_clkdiv	ClockRateControl	Master	-
gtsy_clken	Signal	Master	-
gtsy_clkssel	Value	Master	-
lcp2scp_combined_irq_in[16]	Signal	Slave	-
lcp2scp_mhu_rec_irq_in[16]	Signal	Slave	-
lcp_clkdiv	ClockRateControl	Master	-
lcp_clken	Signal	Master	-
lcp_clkssel	Value	Master	-
m7lockup	Signal	Slave	-
poreset	Signal	Slave	-
pvbus_s	PVBus	Slave	-
scp2lcp_mhu_snd_irq_in[16]	Signal	Slave	-
smcf_trigger_out	Signal	Master	-
sram_pvbus_m	PVBus	Master	-
sram_pvbus_s	PVBus	Slave	-
ss_reset_out	Signal	Master	-
sysresetreq	Signal	Slave	-
watchdog_reset	Signal	Slave	-
ws1_reset_in	Signal	Slave	-
ws1_reset_out	Signal	Master	-
ws1_timer_irq	Signal	Slave	-

Parameters for Infra6_MSCP_PIK

SCP_OR_MCP_CTRL

Type

int

Default value

0x1

Identify this is SCP CTRL or MCP CTRL, SCP = 1, MCP = 0.

diagnostics

Type

int

Default value

0x0

Diagnostics.

ws1_timer.clk_div.div

Type

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

ws1_timer.clk_div.mul

Type

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

ws1_timer.counter.diagnostics

Type

int

Default value

0x2

Diagnostics.

ws1_timer.diagnostics

Type
int
Default value
0x0

Diagnostics.

ws1_timer.timer_interval

Type
int
Default value
0x0

Timer ticks to count before firing interrupt.

3.7.24 Infra6_MSCP_RAS_n_INIT_Ctrl

InfraSysDesign6.x MSCP RAS and INIT control Registers. This model is written in C++.

Iris and MTI instances for Infra6_MSCP_RAS_n_INIT_Ctrl

This model has the following Iris instances:

Table 3-689: Infra6_MSCP_RAS_n_INIT_Ctrl Iris instances

InstanceName	ComponentName
Infra6_MSCP_RAS_n_INIT_Ctrl	Infra6_MSCP_RAS_n_INIT_Ctrl
Infra6_MSCP_RAS_n_INIT_Ctrl.PVBusModifier	PVBusMapper
Infra6_MSCP_RAS_n_INIT_Ctrl.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-690: Infra6_MSCP_RAS_n_INIT_Ctrl MTI instances

InstanceName	ComponentName
Infra6_MSCP_RAS_n_INIT_Ctrl.PVBusModifier	PVBusMapper
Infra6_MSCP_RAS_n_INIT_Ctrl.PVBusSlave	PVBusSlave

Infra6_MSCP_RAS_n_INIT_Ctrl contains the following CADI targets:

- Infra6_MSCP_RAS_n_INIT_Ctrl

Ports for Infra6_MSCP_RAS_n_INIT_Ctrl

Table 3-691: Ports

Name	Protocol	Type	Description
ce_intr_out	Signal	Master	-
cpuwait	Signal	Master	-
dtdcm0_pvbust_m	PVBus	Master	-
dtdcm0_pvbust_s	PVBus	Slave	-
dtdcm1_pvbust_m	PVBus	Master	-
dtdcm1_pvbust_s	PVBus	Slave	-
itcm_pvbust_m	PVBus	Master	-
itcm_pvbust_s	PVBus	Slave	-
mcp_errclear_in	Signal	Slave	-
of_intr_out	Signal	Master	-
pvbust_s	PVBus	Slave	-
ue_intr_out	Signal	Master	-

Parameters for Infra6_MSCP_RAS_n_INIT_Ctrl

diagnostics**Type**

int

Default value

0x0

Diagnostics.

3.7.25 Infra6_SRAM_ECC_RAS

ISD6.x SRAM ECC RAS. This model is written in C++.

Iris and MTI instances for Infra6_SRAM_ECC_RAS

This model has the following Iris instances:

Table 3-692: Infra6_SRAM_ECC_RAS Iris instances

InstanceName	ComponentName
Infra6_SRAM_ECC_RAS	Infra6_SRAM_ECC_RAS
Infra6_SRAM_ECC_RAS.PVBusModifier	PVBusMapper
Infra6_SRAM_ECC_RAS.pvbust_slave	PVBusSlave

This model has the following MTI trace components:

Table 3-693: Infra6_SRAM_ECC_RAS MTI instances

InstanceName	ComponentName
Infra6_SRAM_ECC_RAS.PVBusModifier	PVBusMapper
Infra6_SRAM_ECC_RAS.pvbus_slave	PVBusSlave

Infra6_SRAM_ECC_RAS contains the following CADI targets:

- Infra6_SRAM_ECC_RAS

Ports for Infra6_SRAM_ECC_RAS

Table 3-694: Ports

Name	Protocol	Type	Description
fhi_irq_out	Signal	Master	-
pvbus_m	PVBus	Master	-
pvbus_reg_s	PVBus	Slave	-
pvbus_s	PVBus	Slave	-

Parameters for Infra6_SRAM_ECC_RAS

diagnostics

Type

int

Default value

0x0

Diagnostics.

3.7.26 Infra6_System_PIK

InfraSysDesign6.x System Power Control Kit. This model is written in C++.

Iris and MTI instances for Infra6_System_PIK

This model has the following Iris instances:

Table 3-695: Infra6_System_PIK Iris instances

InstanceName	ComponentName
Infra6_System_PIK	Infra6_System_PIK
Infra6_System_PIK.PPUv1	PPUv1
Infra6_System_PIK.PPUv1.busslave	PVBusSlave
Infra6_System_PIK.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-696: Infra6_System_PIK MTI instances

InstanceName	ComponentName
Infra6_System_PIK.PPUv1	PPUv1
Infra6_System_PIK.PPUv1.busslave	PVBusSlave
Infra6_System_PIK.PVBusSlave	PVBusSlave

Infra6_System_PIK contains the following CADI targets:

- Infra6_System_PIK

Ports for Infra6_System_PIK

Table 3-697: Ports

Name	Protocol	Type	Description
powerdown_out	Signal	Master	-
pvbus_s	PVBus	Slave	-
reset_in	Signal	Slave	-
sys_ppu_irq_out	Signal	Master	-
tcu_10gptsz	Value	Master	-

Parameters for Infra6_System_PIK

diagnostics

Type

int

Default value

0x0

Diagnostics.

systop_pchannel_listener.diagnostics

Type

int

Default value

0x0

Diagnostics.

systop_ppu0.RevD_support

Type

bool

Default value

0x1

Whether to support Rev D locked IRQ.

systop_ppu0.bypass_handshake**Type**

bool

Default value

0x0

Bypass pcsm handshake.

systop_ppu0.dbg_recov**Type**

int

Default value

0x0

Debug Recovery Reset (0: not supported, 1: static mode only, 2: both dynamic & static mode).

systop_ppu0.default_op_dyn_en**Type**

bool

Default value

0x0

Whether to enable operating mode dynamic transition by default.

systop_ppu0.default_op_policy**Type**

int

Default value

0x0

Default operating policy.

systop_ppu0.default_power_state_on**Type**

bool

Default value

0x0

Default power state ON.

systop_ppu0.default_pwr_dyn_en**Type**

bool

Default value

0x0

Whether to enable dynamic power mode transition by default.

systop_ppu0.device_channels**Type**

int

Default value

0x0

Number of device channels (0: P-Channel, 1-8: Q-Channels).

systop_ppu0.dynamic_off**Type**

bool

Default value

0x0

Dynamic Off.

systop_ppu0.dynamic_on**Type**

bool

Default value

0x0

Dynamic On.

systop_ppu0.dynamic_warm_reset**Type**

bool

Default value

0x0

Dynamic Warm Reset.

systop_ppu0.full_ret**Type**

int

Default value

0x0

Full Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

systop_ppu0.func_ret**Type**

int

Default value

0x0

Functional Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

systop_ppu0.is_core_ppu**Type**

bool

Default value

0x0

PPU is core_ppu type which means wake_request would wait till PPU is OFF/OFF_EMU.

systop_ppu0.lock_support**Type**

bool

Default value

0x1

Whether to support OFF lock feature.

systop_ppu0.logic_ret**Type**

int

Default value

0x0

Logic Retention (0: not supported, 1: static, 2: dynamic).

systop_ppu0.mem_off**Type**

int

Default value

0x0

Memory Off (0: not supported, 1: static mode only, 2: both dynamic & static mode).

systop_ppu0.mem_ret**Type**

int

Default value

0x0

Memory Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

systop_ppu0.mem_ret_emu**Type**

int

Default value

0x0

Emulated Memory Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

systop_ppu0.num_opmode_cfg**Type**

int

Default value

0x0

Number of operating modes.

systop_ppu0.off_emu**Type**

int

Default value

0x0

Emulated Off (0: not supported, 1: static mode only, 2: both dynamic & static mode).

systop_ppu0.off_mem_ret_trans_cfg

Type
bool

Default value
0x0

OFF to MEM_RET direct transition configuration (0: not allowed, 1: allowed).

systop_ppu0.op_active_cfg

Type
int

Default value
0x0

Operating mode active configuration (0: Ladder use model, 1: Independent user model).

systop_ppu0.revision

Type
string

Default value
"r1p1"

Revision.

systop_ppu0.use_active_signal

Type
bool

Default value
0x0

Use device-active signal.

3.7.27 Infra6p6_AP_Mgmt_Regs

AP Management Registers Block. This model is written in C++.

Iris and MTI instances for Infra6p6_AP_Mgmt_Regs

This model has the following Iris instances:

Table 3-698: Infra6p6_AP_Mgmt_Regs Iris instances

InstanceName	ComponentName
Infra6p6_AP_Mgmt_Regs	Infra6p6_AP_Mgmt_Regs

InstanceName	ComponentName
Infra6p6_AP_Mgmt_Regs.apb	PVBusSlave

This model has the following MTI trace components:

Table 3-699: Infra6p6_AP_Mgmt_Regs MTI instances

InstanceName	ComponentName
Infra6p6_AP_Mgmt_Regs.apb	PVBusSlave

Infra6p6_AP_Mgmt_Regs contains the following CADI targets:

- Infra6p6_AP_Mgmt_Regs

Ports for Infra6p6_AP_Mgmt_Regs

Table 3-700: Ports

Name	Protocol	Type	Description
ap_cons_irq_in[32]	Signal	Slave	AP Consolidated interrupts
ap_cons_irq_out_lcp[32]	Signal	Master	AP Consolidated interrupts to LCP
ap_cons_irq_out_scp[32]	Signal	Master	AP Consolidated interrupts to SCP
ap_warm_reset_req	Signal	Master	AP Warm Reset Request
apb	PVBus	Slave	APB Subordinate Interface - Access to registers
mgi_irq_in0[32]	Signal	Slave	SMCF MGI interrupts indexed [0..31]
mgi_irq_in1[32]	Signal	Slave	SMCF MGI interrupts indexed [32..63]
mgi_irq_out0_lcp[32]	Signal	Master	MGI interrupt indexed [0..31] to LCP
mgi_irq_out0_scp[32]	Signal	Master	MGI interrupts indexed [0..31] to SCP
mgi_irq_out1_lcp[32]	Signal	Master	MGI interrupt indexed [32..63] to LCP
mgi_irq_out1_scp[32]	Signal	Master	MGI interrupts indexed [32..63] to SCP
reset_in	Signal	Slave	Reset in signal

Parameters for Infra6p6_AP_Mgmt_Regs

AP_GRP_ACCESS_ID

Type

int

Default value

0x0

AP Group ID in the SoC.

LCP_GRP_ACCESS_ID

Type

int

Default value

0x0

LCP Group ID associated with this AP Group.

diagnostics**Type**

int

Default value

0x2

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.7.28 Infra6p6_LCP_Extended_Ctrl_Regs

LCP Subsystem Control Register Block. This model is written in C++.

Iris and MTI instances for Infra6p6_LCP_Extended_Ctrl_Regs

This model has the following Iris instances:

Table 3-701: Infra6p6_LCP_Extended_Ctrl_Regs Iris instances

InstanceName	ComponentName
Infra6p6_LCP_Extended_Ctrl_Regs	Infra6p6_LCP_Extended_Ctrl_Regs
Infra6p6_LCP_Extended_Ctrl_Regs.ApbPVBUSubordinate	PVBusSlave

This model has the following MTI trace components:

Table 3-702: Infra6p6_LCP_Extended_Ctrl_Regs MTI instances

InstanceName	ComponentName
Infra6p6_LCP_Extended_Ctrl_Regs.ApbPVBUSubordinate	PVBusSlave

Infra6p6_LCP_Extended_Ctrl_Regs contains the following CADI targets:

- Infra6p6_LCP_Extended_Ctrl_Regs

Ports for Infra6p6_LCP_Extended_Ctrl_Regs

Table 3-703: Ports

Name	Protocol	Type	Description
ap2lcp_mhu_irq_ctrl	StateSignal	Slave	AP to LCP MHU Combined interrupt Control
ap_cons_irq_ctrl	StateSignal	Slave	Consolidated AP interrupts Control
ap_core_ppu_irq_ctrl	StateSignal	Slave	Consolidated AP Core PPU interrupts Control
ap_mgi_irq_ctrl	StateSignal	Slave	MGI Interrupts Control
apb	PVBus	Slave	APB Subordinate Interface - Access to registers

Name	Protocol	Type	Description
lcp_tcm_acq_en	StateSignal	Slave	LCP Core TCMs Control
reset_in	Signal	Slave	Reset in signal
smcf_trigger_in_scp	Signal	Slave	SMCF trigger driven by the SCP
smcf_trigger_out	Signal	Master	SMCF trigger out signal
warm_reset_req	Signal	Slave	AP Warm Reset Request

Parameters for Infra6p6_LCP_Extended_Ctrl_Regs

LCP_CTRL_RATIO

Type

int

Default value

0x0

Number of APs that share an LCP. The LCP is shared between LCP_CTRL_RATIO + 1 APs.

LCP_GRP_ACCESS_ID

Type

int

Default value

0x0

LCP Access region ID in the SoC.

diagnostics

Type

int

Default value

0x2

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.7.29 Infra6p6_LCP_External_Ctrl_Regs

LCP Control Registers Block. This model is written in C++.

Iris and MTI instances for Infra6p6_LCP_External_Ctrl_Regs

This model has the following Iris instances:

Table 3-704: Infra6p6_LCP_External_Ctrl_Regs Iris instances

InstanceName	ComponentName
Infra6p6_LCP_External_Ctrl_Regs	Infra6p6_LCP_External_Ctrl_Regs
Infra6p6_LCP_External_Ctrl_Regs.ApbPVBUSubordinate	PVBusSlave
Infra6p6_LCP_External_Ctrl_Regs.PVBusMapper	PVBusMapper

This model has the following MTI trace components:

Table 3-705: Infra6p6_LCP_External_Ctrl_Regs MTI instances

InstanceName	ComponentName
Infra6p6_LCP_External_Ctrl_Regs.ApbPVBUSubordinate	PVBusSlave
Infra6p6_LCP_External_Ctrl_Regs.PVBusMapper	PVBusMapper

Infra6p6_LCP_External_Ctrl_Regs contains the following CADI targets:

- Infra6p6_LCP_External_Ctrl_Regs

Ports for Infra6p6_LCP_External_Ctrl_Regs

Table 3-706: Ports

Name	Protocol	Type	Description
access_status_irq_out	Signal	Master	LCP Access Status Interrupt
ap_cons_irq_in[32]	Signal	Slave	Consolidated interrupts from the APs [0..SHARED_AP_NUM-1]
ap_cons_irq_out_scp	Signal	Master	Consolidated AP core IRQ to SCP
ap_core_ppu_irq_in[32]	Signal	Slave	Interrupts from the AP core PPU [0..SHARED_AP_NUM-1]
ap_core_ppu_irq_out_scp	Signal	Master	Consolidated AP PPU IRQ to SCP
ap_mgi_cons_irq_in[32]	Signal	Slave	SMCF MGI Consolidated interrupts [0..SHARED_AP_NUM-1]
ap_mgi_cons_irq_out_scp	Signal	Master	Consolidated MGI IRQ to SCP
apb	PVBus	Slave	APB Subordinate Interface - Access to registers
combined_irq_in[32]	Signal	Slave	LCP Combined interrupt
combined_irq_out_scp	Signal	Master	Consolidated LCP IRQ to SCP
reset_in	Signal	Slave	Reset in signal
uart_filter_pvbus_m	PVBus	Master	-
uart_filter_pvbus_s	PVBus	Slave	Ports to check if LCP UART are accessible
uart_irq_in	Signal	Slave	LCP UART interrupt
uart_irq_out	Signal	Master	LCP UART interrupt is routed to this LCP
warm_reset_req	Signal	Master	Warm Reset Request

Parameters for Infra6p6_LCP_External_Ctrl_Regs

SHARED_AP_NUM

Type

int

Default value

0x20

Number of APs that share an LCP.

diagnostics**Type**

int

Default value

0x2

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.7.30 IntelStrataFlashJ3

Intel Strata Flash J3 model. This model is written in C++.

Iris and MTI instances for IntelStrataFlashJ3

This model has the following Iris instances:

Table 3-707: IntelStrataFlashJ3 Iris instances

InstanceName	ComponentName
IntelStrataFlashJ3	IntelStrataFlashJ3
IntelStrataFlashJ3.PVBusMapper	PVBusMapper
IntelStrataFlashJ3.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-708: IntelStrataFlashJ3 MTI instances

InstanceName	ComponentName
IntelStrataFlashJ3.PVBusMapper	PVBusMapper
IntelStrataFlashJ3.PVBusSlave	PVBusSlave

IntelStrataFlashJ3 contains the following CADI targets:

- IntelStrataFlashJ3

Ports for IntelStrataFlashJ3**Table 3-709: Ports**

Name	Protocol	Type	Description
flashloader	FlashLoaderPort	Slave	-
mem_port	PVDevice	Slave	-
pdbus	PVBus	Slave	-

Parameters for IntelStrataFlashJ3

diagnostics

Type

int

Default value

0x0

Diagnostic level.

enable_read_status_logic

Type

bool

Default value

0x0

Enables logic to handle the status register reads as per the '3 Volt Intel StrataFlash Memory' specification.

model_blocklock

Type

bool

Default value

0x0

Model per-block locking and set all the blocks to locked state on reset.

size

Type

int

Default value

0x40000

Memory Size.

trapwrite

Type

bool

Default value

0x0

Generate abort on write.

unphysical_writes**Type**

bool

Default value

0x1

Writes to flash are overwrite not AND.

3.7.31 Interrupt_Router

Interrupt Router Registers. This model is written in C++.

Ports for Interrupt_Router

Table 3-710: 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	-

Number of Interrupt Controllers Interrupt interface.

num_shd_int

Type
int

Default value
0x1

Number of shared interrupts supported.

ro_access

Type
int

Default value
0x0

Stream ID of master.

rw_access

Type
int

Default value
0x0

Stream ID of master with Read Write access.

3.7.32 **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 3-711: IoTSS3_ManagerSecurityController Iris instances

InstanceName	ComponentName
IoTSS3_ManagerSecurityController	IoTSS3_ManagerSecurityController
IoTSS3_ManagerSecurityController.PVBusModifier	PVBusMapper

This model has the following MTI trace components:

Table 3-712: IoTSS3_ManagerSecurityController MTI instances

InstanceName	ComponentName
IoTSS3_ManagerSecurityController.PVBusModifier	PVBusMapper

IoTSS3_ManagerSecurityController contains the following CADI targets:

- IoTSS3_ManagerSecurityController

Ports for IoTSS3_ManagerSecurityController

Table 3-713: Ports

Name	Protocol	Type	Description
cfg_nonsec	ValueState	Slave	-
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

Interrupt enable read.

IRQ_ENABLE_WR

Type

bool

Default value

0x1

Interrupt enable write.

diagnostics

Type

int

Default value

0x0

Diagnostics.

3.7.33 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 3-714: IoTSS3_MemoryProtectionController Iris instances

InstanceName	ComponentName
IoTSS3_MemoryProtectionController	IoTSS3_MemoryProtectionController
IoTSS3_MemoryProtectionController.PVBusModifier	PVBusMapper
IoTSS3_MemoryProtectionController.PVBusSlave	PVBusSlave
IoTSS3_MemoryProtectionController.SchedulerThreadEvent	SchedulerThreadEvent

This model has the following MTI trace components:

Table 3-715: IoTSS3_MemoryProtectionController MTI instances

InstanceName	ComponentName
IoTSS3_MemoryProtectionController.PVBusModifier	PVBusMapper
IoTSS3_MemoryProtectionController.PVBusSlave	PVBusSlave

IoTSS3_MemoryProtectionController contains the following CADI targets:

- IoTSS3_MemoryProtectionController

Ports for IoTSS3_MemoryProtectionController

Table 3-716: 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

Maximum block index configuration.

BLK_SIZE**Type**

int

Default value

0x3

Block size configuration.

GATE_PRESENT**Type**

bool

Default value

0x1

Memory gating logic present/not.

IRQ_ENABLE_RD**Type**

bool

Default value

0x1

Interrupt enable read.

IRQ_ENABLE_WR**Type**

bool

Default value

0x1

Interrupt enable write.

diagnostics**Type**

int

Default value

0x0

Diagnostics.

3.7.34 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 3-717: IoTSS3_SecureAccessConfig Iris instances

InstanceName	ComponentName
IoTSS3_SecureAccessConfig	IoTSS3_SecureAccessConfig
IoTSS3_SecureAccessConfig.PVBusMaster	PVBusMaster
IoTSS3_SecureAccessConfig.PVBusModifier	PVBusMapper
IoTSS3_SecureAccessConfig.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-718: IoTSS3_SecureAccessConfig MTI instances

InstanceName	ComponentName
IoTSS3_SecureAccessConfig.PVBusMaster	PVBusMaster
IoTSS3_SecureAccessConfig.PVBusModifier	PVBusMapper
IoTSS3_SecureAccessConfig.PVBusSlave	PVBusSlave

IoTSS3_SecureAccessConfig contains the following CADI targets:

- IoTSS3_SecureAccessConfig

Ports for IoTSS3_SecureAccessConfig

Table 3-719: 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	-
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	-

Name	Protocol	Type	Description
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 IoTSS3_SecureAccessConfig

CODENSC

Type

bool

Default value

0x0

Whether 0x10000000..0x1FFFFFFF is non-secure-callable.

DISABLE_GATING

Type

bool

Default value

0x0

Disable Memory gating logic.

IGNORE_MEM_MAP

Type

bool

Default value

0x0

Ignore Memory mapping logic.

MAINPPCEXP_DIS0**Type**

int

Default value

0x0

Disables support for individual bits on the MAINNSPPCEXP0 and MAINPPPCEXP0 buses.

MAINPPCEXP_DIS1**Type**

int

Default value

0x0

Disables support for individual bits on the MAINNSPPCEXP1 and MAINPPPCEXP1 buses.

MAINPPCEXP_DIS2**Type**

int

Default value

0x0

Disables support for individual bits on the MAINNSPPCEXP2 and MAINPPPCEXP2 buses.

MAINPPCEXP_DIS3**Type**

int

Default value

0x0

Disables support for individual bits on the MAINNSPPCEXP3 and MAINPPPCEXP3 buses.

PERIPHPPCEXP_DIS0**Type**

int

Default value

0x0

Disables support for individual bits on the PERIPHNSPPCEXP0 and PERIPHPPPCEXP0 buses.

PERIPHPPCEXP_DIS1

Type

int

Default value

0x0

Disables support for individual bits on the PERIPHNSPPCEXP1 and PERIPHPPPCEXP1 buses.

PERIPHPPCEXP_DIS2

Type

int

Default value

0x0

Disables support for individual bits on the PERIPHNSPPCEXP2 and PERIPHPPPCEXP2 buses.

PERIPHPPCEXP_DIS3

Type

int

Default value

0x0

Disables support for individual bits on the PERIPHNSPPCEXP3 and PERIPHPPPCEXP3 buses.

RAMNSC

Type

bool

Default value

0x0

Whether 0x30000000..0x3FFFFFFF is non-secure-callable.

diagnostics

Type

int

Default value

0x0

Diagnostics.

3.7.35 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 3-720: Juno_sysregs Iris instances

InstanceName	ComponentName
Juno_sysregs	Juno_sysregs
Juno_sysregs.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-721: Juno_sysregs MTI instances

InstanceName	ComponentName
Juno_sysregs.PVBusSlave	PVBusSlave

Juno_sysregs contains the following CADI targets:

- Juno_sysregs

Ports for Juno_sysregs

Table 3-722: 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

Diagnostics.

pcie_mac
Type
int
Default value
0x2f7000001

PCIe MAC address.

rev
Type
int
Default value
0x0

Board revision.

3.7.36 LS64TestingFIFO

FIFO peripheral supporting LS64 accesses for testing purposes. This model is written in LISA+.

Iris and MTI instances for LS64TestingFIFO

This model has the following Iris instances:

Table 3-723: LS64TestingFIFO Iris instances

InstanceName	ComponentName
LS64TestingFIFO	LS64TestingFIFO
LS64TestingFIFO.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Table 3-724: LS64TestingFIFO MTI instances

InstanceName	ComponentName
LS64TestingFIFO.pvbusslave	PVBusSlave

LS64TestingFIFO contains the following CADI targets:

- LS64TestingFIFO

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.

Ports for LS64TestingFIFO

Table 3-725: Ports

Name	Protocol	Type	Description
pvbuss_s	PVBus	Slave	Bus subordinate interface.

Parameters for LS64TestingFIFO

buffer_size

Type

int

Default value

0x2

The number of 64-byte slots in the FIFO.

op_type

Type

int

Default value

0x1

The operation performed on the 64-byte transaction data 0 - None, 1 - Bitwise Negate.

3.7.37 MemoryElement_TZC400_cpp

Memory Element with TZC-400. This model is written in C++.

Iris and MTI instances for MemoryElement_TZC400_cpp

This model has the following Iris instances:

Table 3-726: MemoryElement_TZC400_cpp Iris instances

InstanceName	ComponentName
MemoryElement_TZC400_cpp	MemoryElement_TZC400
MemoryElement_TZC400_cpp.TZC_400	TZC_400
MemoryElement_TZC400_cpp.TZC_400.apbslave[0]	PVBusSlave
MemoryElement_TZC400_cpp.TZC_400.filter0	filter0
MemoryElement_TZC400_cpp.TZC_400.filter0.BusMapper	PVBusMapper
MemoryElement_TZC400_cpp.TZC_400.filter1	filter1
MemoryElement_TZC400_cpp.TZC_400.filter1.BusMapper	PVBusMapper

InstanceName	ComponentName
MemoryElement_TZC400_cpp.TZC_400.filter2	filter2
MemoryElement_TZC400_cpp.TZC_400.filter2.BusMapper	PVBusMapper
MemoryElement_TZC400_cpp.TZC_400.filter3	filter3
MemoryElement_TZC400_cpp.TZC_400.filter3.BusMapper	PVBusMapper
MemoryElement_TZC400_cpp.TZFilterUnit	ExportTest.MemoryElement_TZC400_cppTZFilterUnit
MemoryElement_TZC400_cpp.TZFilterUnit.BusMapper	PVBusMapper

This model has the following MTI trace components:

Table 3-727: MemoryElement_TZC400_cpp MTI instances

InstanceName	ComponentName
MemoryElement_TZC400_cpp.TZC_400.apbslave[0]	PVBusSlave
MemoryElement_TZC400_cpp.TZC_400.filter0.BusMapper	PVBusMapper
MemoryElement_TZC400_cpp.TZC_400.filter1.BusMapper	PVBusMapper
MemoryElement_TZC400_cpp.TZC_400.filter2.BusMapper	PVBusMapper
MemoryElement_TZC400_cpp.TZC_400.filter3.BusMapper	PVBusMapper
MemoryElement_TZC400_cpp.TZFilterUnit.BusMapper	PVBusMapper

MemoryElement_TZC400_cpp contains the following CADI targets:

- MemoryElement_TZC400

Ports for MemoryElement_TZC400_cpp

Table 3-728: Ports

Name	Protocol	Type	Description
tzc_apb_pvbus_s	PVBus	Slave	-
tzc_filter_pvbus_m[4]	PVBus	Master	-
tzc_filter_pvbus_s[4]	PVBus	Slave	-
tzc_irq	Signal	Master	-

Parameters for MemoryElement_TZC400_cpp

addr_width

Type

int

Default value

0x40

Address width: Supported widths are 32, 36, 40, 48, 64 bits and all other values are reserved.

num_filter

Type

int

Default value

0x4

Number of filters in each TZC. Supported values are 1, 2, 4. 3 is Reserved.

num_region

Type

int

Default value

0x9

Number of regions.

3.7.38 MemoryMappedGenericTimer

ARM Generic Timer. This model is written in C++.

Iris and MTI instances for MemoryMappedGenericTimer

This model has the following Iris instances:

Table 3-729: MemoryMappedGenericTimer Iris instances

InstanceName	ComponentName
MemoryMappedGenericTimer	MemoryMappedGenericTimer
MemoryMappedGenericTimer.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-730: MemoryMappedGenericTimer MTI instances

InstanceName	ComponentName
MemoryMappedGenericTimer.PVBusSlave	PVBusSlave

MemoryMappedGenericTimer contains the following CADI targets:

- MemoryMappedGenericTimer

Ports for MemoryMappedGenericTimer

Table 3-731: Ports

Name	Protocol	Type	Description
cntpsirq[8]	Signal	Master	-
cntvalueb	CounterInterface	Slave	-

Name	Protocol	Type	Description
pvbuse_base_s[8]	PVBus	Slave	-
pvbuse_ctlbase_s	PVBus	Slave	-
pvbuse_el0base_s[8]	PVBus	Slave	-
timer_reset	Signal	Slave	-

Parameters for MemoryMappedGenericTimer

bypass_ctlbase

Type

bool

Default value

0x0

Bypass CNTBase Access Control. Enable if only timer frame feature is required without CNTBase access control.

cntel0acr_implemented

Type

int

Default value

0x0

A bit-field of 8 bits, where bit {n} enables CNTELOACR for timer frame {n}.

diagnostics

Type

int

Default value

0x0

Diagnostics.

frame_security

Type

string

Default value

""

Hard-wired/configurable security for frames (N/S/X, one character per timer frame).

num_timers**Type**

int

Default value

0x1

Number of timer frames.

3.7.39 MemoryMappedGenericWatchdog

ARM Generic Watchdog. This model is written in C++.

Iris and MTI instances for MemoryMappedGenericWatchdog

This model has the following Iris instances:

Table 3-732: MemoryMappedGenericWatchdog Iris instances

InstanceName	ComponentName
MemoryMappedGenericWatchdog	MemoryMappedGenericWatchdog
MemoryMappedGenericWatchdog.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-733: MemoryMappedGenericWatchdog MTI instances

InstanceName	ComponentName
MemoryMappedGenericWatchdog.PVBusSlave	PVBusSlave

MemoryMappedGenericWatchdog contains the following CADI targets:

- MemoryMappedGenericWatchdog

About MemoryMappedGenericWatchdog

This is a high-level watchdog that generates two interrupts rather than an interrupt followed by a reset.

Ports for MemoryMappedGenericWatchdog

Table 3-734: 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

Non-Secure.

arch_version

Type

int

Default value

0x0

Architecture version. Available 0 and 1.

diagnostics

Type

int

Default value

0x0

Diagnostics.

product_id

Type

int

Default value

0x0

Product Identifier.

3.7.40 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 3-735: NonVolatileCounter Iris instances

InstanceName	ComponentName
NonVolatileCounter	NonVolatileCounter
NonVolatileCounter.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-736: NonVolatileCounter MTI instances

InstanceName	ComponentName
NonVolatileCounter	NonVolatileCounter
NonVolatileCounter.PVBusSlave	PVBusSlave

NonVolatileCounter contains the following CADI targets:

- NonVolatileCounter

Ports for NonVolatileCounter

Table 3-737: Ports

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	-

Parameters for NonVolatileCounter

diagnostics

Type

int

Default value

0x0

Diagnostics.

rst_non_tz_fw_cnt

Type

int

Default value

0x0

Value of NON_TZ_FW_CNT at reset.

rst_tz_fw_cnt

Type

int

Default value

0x0

Value of TZ_FW_CNT at reset.

secure

Type

int

Default value

0x1

Instantiate model as Secure (1) or NS (0).

version

Type

string

Default value

"r0"

Version of the model functionality. Valid values are r0 and r1.

3.7.41 PCIeATC

This model is written in C++.

Iris and MTI instances for PCIeATC

This model has the following Iris instances:

Table 3-738: 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 3-739: PCIeATC MTI instances

InstanceName	ComponentName
PCIeATC	atc
PCIeATC.ExportTest.PCIeATC.mapper	PVBusMapper
PCIeATC.pvbus_id_routed_s[0]	PVBusSlave

PCIeATC contains the following CADI targets:

- validation_atc

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()`.

Ports for PCIeATC

Table 3-740: 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

The maximum number of ATC entries. 0 is effectively a large number.

seed

Type

int

Default value

0x12345678

Seed for a random number generator.

3.7.42 PLLClockControl

Clock Rate Control. This model is written in C++.

Iris and MTI instances for PLLClockControl

This model has the following Iris instances:

Table 3-741: PLLClockControl Iris instances

InstanceName	ComponentName
PLLClockControl	PLLClockControl
PLLClockControl.ClockDivider	ClockDivider

This model has the following MTI trace components:

Table 3-742: PLLClockControl MTI instances

InstanceName	ComponentName
PLLClockControl.ClockDivider	ClockDivider

PLLClockControl contains the following CADI targets:

- PLLClockControl

Ports for PLLClockControl

Table 3-743: 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	-
lock	Signal	Master	-
refclk_in	ClockSignal	Slave	-
unlock	Signal	Master	-

Parameters for PLLClockControl

diagnostics

Type

int

Default value

0x0

Diagnostics.

3.7.43 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 3-744: 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

Multi-threaded mode.

3.7.44 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 3-745: 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

PS2Keyboard contains the following CADI targets:

- ClockTimerThread

- ClockTimerThread64
- PS2Keyboard
- SchedulerThread
- SchedulerThreadEvent

Ports for PS2Keyboard

Table 3-746: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock signal to rate at which PS2Data signals are generated.
keyboard	KeyboardStatus	Slave	Receives keyboard input from, for example, the Visualisation component.
ps2	PS2Data	Master	Connection to the PS/2 controller, for example, the PL050_KMI.

3.7.45 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 LISA+.

Iris and MTI instances for PS2Mouse

This model has the following Iris instances:

Table 3-747: PS2Mouse Iris instances

InstanceName	ComponentName
PS2Mouse	PS2Mouse
PS2Mouse.ps2_clocktimer	ClockTimerThread
PS2Mouse.ps2_clocktimer.timer	ClockTimerThread64
PS2Mouse.ps2_clocktimer.timer.thread	SchedulerThread
PS2Mouse.ps2_clocktimer.timer.thread_event	SchedulerThreadEvent

PS2Mouse contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PS2Mouse
- SchedulerThread
- SchedulerThreadEvent

Ports for PS2Mouse

Table 3-748: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock signal to rate at which PS2Data signals are generated.

Name	Protocol	Type	Description
mouse	MouseStatus	Slave	Receives keyboard input from, for example, the Visualisation component.
ps2	PS2Data	Master	Connection to the PS/2 controller, for example the PL050_KMI.

3.7.46 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 3-749: PVBusGICv3Comms Iris instances

InstanceName	ComponentName
PVBusGICv3Comms	PVBusGICv3Comms
PVBusGICv3Comms.bus_slave	PVBusSlave

This model has the following MTI trace components:

Table 3-750: PVBusGICv3Comms MTI instances

InstanceName	ComponentName
PVBusGICv3Comms	PVBusGICv3Comms
PVBusGICv3Comms.bus_slave	PVBusSlave

PVBusGICv3Comms contains the following CADI targets:

- [PVBusGICv3Comms](#)

Ports for PVBusGICv3Comms

Table 3-751: Ports

Name	Protocol	Type	Description
axi_master_id_m[256]	Value	Master	-
distributor_m[256]	GICv3Comms	Master	-
pvbus_m	PVBus	Master	-
pvbus_s	PVBus	Slave	-

3.7.47 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 file defines 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 entity is represented by `MetaDataController`. This 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 will drastically slow 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`.

Ports for PVMetaDataController

Table 3-752: Ports

Name	Protocol	Type	Description
<code>pvbuss_m</code>	PVBUS	Master	-
<code>pvbuss_s</code>	PVBUS	Slave	-

Parameters for PVMetaDataController

`init_value`

Type

int

Default value

0xd

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

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 ≥ 64 bytes and a power of 2, defaulting to 4KiB. The maximum block size supported is 4KiB. The carveout region size must be ≥ 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 },] }.

mte_tag_carveout_json_file

Type
string

Default value
""

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. .

mte_tag_carveout_tag_order

Type
string

Default value
"little-endian"

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. THIS PARAMETER HAS NO FUNCTIONALITY AT THE MOMENT.

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.

3.7.48 PchannelListener

Provides a dummy PChannel device to accept all request. This model is written in C++.

Ports for PchannelListener

Table 3-753: Ports

Name	Protocol	Type	Description
dev_pchannel_s	PChannel	Slave	-

Parameters for PchannelListener

diagnostics

Type
int

Default value
0x0

Diagnostics.

3.7.49 RAMDevice

RAM device, can be dynamic or static ram. This model is written in LISA+.

Iris and MTI instances for RAMDevice

This model has the following Iris instances:

Table 3-754: RAMDevice Iris instances

InstanceName	ComponentName
RAMDevice	RAMDevice
RAMDevice.bus_slave	PVBusSlave

This model has the following MTI trace components:

Table 3-755: RAMDevice MTI instances

InstanceName	ComponentName
RAMDevice	RAMDevice
RAMDevice.bus_slave	PVBusSlave

RAMDevice contains the following CADI targets:

- RAMDevice

About RAMDevice

As a generic device, this component does not have a hardware revision code.

Ports for RAMDevice

Table 3-756: Ports

Name	Protocol	Type	Description
pvbus	PVBus	Slave	Bus slave interface.

Parameters for RAMDevice

enable_atomic_ops

Type

bool

Default value

0x0

Supports Atomic Operations.

fill1

Type

int

Default value

0x0

Fill pattern 1, initialise memory at start of simulation with alternating fill1, fill2 pattern.

fill2

Type

int

Default value

0x0

Fill pattern 2, initialise memory at start of simulation with alternating fill1, fill2 pattern.

read_latency

Type

int

Default value

0x0

Memory read latency (ps/byte).

size

Type

int

Default value

0x100000000

Memory Size.

write_latency

Type

int

Default value

0x0

Memory write latency (ps/byte).

3.7.50 RAMDevice_cpp

RAM device, can be dynamic or static ram. This model is written in C++.

Iris and MTI instances for RAMDevice_cpp

This model has the following Iris instances:

Table 3-757: RAMDevice_cpp Iris instances

InstanceName	ComponentName
RAMDevice_cpp	RAMDevice
RAMDevice_cpp.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-758: RAMDevice_cpp MTI instances

InstanceName	ComponentName
RAMDevice_cpp	RAMDevice
RAMDevice_cpp.PVBusSlave	PVBusSlave

RAMDevice_cpp contains the following CADI targets:

- RAMDevice

Ports for RAMDevice_cpp

Table 3-759: Ports

Name	Protocol	Type	Description
pvbus	PVBus	Slave	-

Parameters for RAMDevice_cpp

enable_atomic_ops

Type

bool

Default value

0x0

Supports Atomic Operations.

fill1**Type**

int

Default value

0xdfdfdfcf

Fill pattern 1, initialise memory at start of simulation with alternating fill1, fill2 pattern.

fill2**Type**

int

Default value

0xcfdfdfdf

Fill pattern 2, initialise memory at start of simulation with alternating fill1, fill2 pattern.

read_latency**Type**

int

Default value

0x0

Memory read latency (ps/byte).

size**Type**

int

Default value

0x100000000

Memory Size.

write_latency**Type**

int

Default value

0x0

Memory write latency (ps/byte).

3.7.51 ROM

Simple ROM device. This model is written in LISA+.

Iris and MTI instances for ROM

This model has the following Iris instances:

Table 3-760: ROM Iris instances

InstanceName	ComponentName
ROM	ROM
ROM.bus_slave	PVBusSlave

This model has the following MTI trace components:

Table 3-761: ROM MTI instances

InstanceName	ComponentName
ROM.bus_slave	PVBusSlave

ROM contains the following CADI targets:

- ROM

Ports for ROM

Table 3-762: Ports

Name	Protocol	Type	Description
pvbuss	PVBus	Slave	-

Parameters for ROM

abort_writes

Type

bool

Default value

0x0

Abort writes instead of ignoring them.

log2_size

Type

int

Default value

0x14

Log2 size (bytes) e.g. 20 is 1 MiB.

raw_image**Type**

string

Default value

""

Raw image file to load at init time.

3.7.52 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 3-763: 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 3-764: RSE_Integ_Regs MTI instances

InstanceName	ComponentName
RSE_Integ_Regs.ClockDivider	ClockDivider
RSE_Integ_Regs.PVBusSlave	PVBusSlave

RSE_Integ_Regs contains the following CADI targets:

- [RSE_Integration_Registers](#)

Ports for RSE_Integ_Regs

Table 3-765: Ports

Name	Protocol	Type	Description
EXTMCPRESETn	Signal	Master	-
EXTTSCPRESETn	Signal	Master	-
MCP_ATU_AP	Signal	Master	-
MCP_RAS_ERR_CLEAR	Signal	Master	-
pvbus_s	PVBus	Slave	-
REFCLK	ClockSignal	Slave	-
reset_in	Signal	Slave	-

Name	Protocol	Type	Description
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

RSE Integration Register Chip ID.

diagnostics

Type

int

Default value

0x0

RSE Integration Registers Diagnostics.

multichip_mode

Type

bool

Default value

0x0

RSE Integration Register multichip mode.

3.7.53 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 3-766: RandomNumberGenerator Iris instances

InstanceName	ComponentName
RandomNumberGenerator	RandomNumberGenerator

InstanceName	ComponentName
RandomNumberGenerator.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-767: RandomNumberGenerator MTI instances

InstanceName	ComponentName
RandomNumberGenerator.PVBusSlave	PVBusSlave

RandomNumberGenerator contains the following CADI targets:

- RandomNumberGenerator

Ports for RandomNumberGenerator

Table 3-768: 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

Diagnostics.

seed

Type

int

Default value

0x0

Random number seed.

3.7.54 RealTimeLimiter

Real Time Limiter. This model is written in LISA+.

Iris and MTI instances for RealTimeLimiter

This model has the following Iris instances:

Table 3-769: RealTimeLimiter Iris instances

InstanceName	ComponentName
RealTimeLimiter	RealTimeLimiter
RealTimeLimiter.divider	ClockDivider

This model has the following MTI trace components:

Table 3-770: RealTimeLimiter MTI instances

InstanceName	ComponentName
RealTimeLimiter.divider	ClockDivider

RealTimeLimiter contains the following CADI targets:

- ClockDivider
- RealTimeLimiter

Ports for RealTimeLimiter

Table 3-771: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock input.

Parameters for RealTimeLimiter

ENABLE

Type

bool

Default value

0x0

Rate limit simulation.

RELATIVE_SPEED

Type

int

Default value

0x64

Rate limit to at most this percentage of real time (100: limit to wall clock rate).

divider.div

Type

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

3.7.55 RealtimeClockTimer

Host Time Based Timer Module for Generic Timers. This model is written in C++.

Ports for RealtimeClockTimer**Table 3-772: Ports**

Name	Protocol	Type	Description
set_frequency	Value_64	Slave	-
timer_callback	TimerCallback64	Master	-
timer_control	TimerControl64	Slave	-

3.7.56 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 3-773: 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 3-774: RemapDecoder MTI instances

InstanceName	ComponentName
RemapDecoder.bus_switch.pvbus_mapper	PVBusMapper

RemapDecoder contains the following CADI targets:

- RemapDecoder
- TZSwitch

Ports for RemapDecoder

Table 3-775: 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

Normal Port.

`bus_switch.secure`

Type

int

Default value

0x1

Secure Port.

3.7.57 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 3-776: RootKeyStorage Iris instances

InstanceName	ComponentName
RootKeyStorage	RootKeyStorage
RootKeyStorage.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-777: RootKeyStorage MTI instances

InstanceName	ComponentName
RootKeyStorage.PVBusSlave	PVBusSlave

RootKeyStorage contains the following CADI targets:

- RootKeyStorage

Ports for RootKeyStorage

Table 3-778: Ports

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	-

Parameters for RootKeyStorage

diagnostics

Type

int

Default value

0x0

Diagnostics.

hw_unique_key

Type

string

Default value

"00000000 00000000 00000000 00000000"

Hardware Unique Key (128-bit, 4 std::hex words).

hw_unique_key_hex

Type

string

Default value

""

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"

Private Endorsement Key (256-bit, 8 std::hex words).

private_key_hex**Type**

string

Default value

""

Private Key (256-bit, little-endian std::hex byte stream).

public_key**Type**

string

Default value

"00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000"

Public Key (256-bit, 8 std::hex words).

public_key_hex**Type**

string

Default value

""

Public Key (256-bit, little-endian std::hex byte stream).

ss_key**Type**

string

Default value

"00000000 00000000 00000000 00000000"

Secret Symmetric Key (128-bit, 4 std::hex words).

ss_key_hex**Type**

string

Default value

""

Secret Symmetric Key (128-bit, little-endian std::hex byte stream).

version

Type

string

Default value

"r1"

Version of the model functionality. Valid values are r0 and r1.

3.7.58 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 3-779: 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 3-780: SI_System_Ctrl_Regs MTI instances

InstanceName	ComponentName
SI_System_Ctrl_Regs.PVBusSlave	PVBusSlave

SI_System_Ctrl_Regs contains the following CADI targets:

- SI_System_Control_Registers

Ports for SI_System_Ctrl_Regs

Table 3-781: 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

CL0_C0_CFGRVBARADDR.

c11_c0_cfgrvbaraddr

Type

int

Default value

0x140000000

CL1_C0_CFGRVBARADDR.

c11_c1_cfgrvbaraddr

Type

int

Default value

0x140002000

CL1_C1_CFGRVBARADDR.

c12_c0_cfgrvbaraddr

Type

int

Default value

0x160000000

CL2_C0_CFGRVBARADDR.

c12_c1_cfgrvbaraddr

Type

int

Default value

0x160002000

CL2_C1_CFGRVBARADDR.

cl2_c2_cfgrvbaraddr**Type**

int

Default value

0x160004000

CL2_C2_CFGRVBARADDR.

cl2_c3_cfgrvbaraddr**Type**

int

Default value

0x160008000

CL2_C3_CFGRVBARADDR.

cpuhalt_reset**Type**

int

Default value

0x0

CPU HALT Reset value.

diagnostics**Type**

int

Default value

0x0

Diagnostics.

3.7.59 SMCF

System Monitoring Control Framework (SMCF). This model is written in C++.

Changes in 11.27.17

Ports added:

- mli_hsp_enable_ack_in
- mli_hsp_enable_out
- smcf_mli_pvbus_m

Ports removed:

- mli_amu_pvbus_m

Iris and MTI instances for SMCF

This model has the following Iris instances:

Table 3-782: SMCF Iris instances

InstanceName	ComponentName
SMCF	SMCF

SMCF contains the following CADI targets:

- SMCF

Ports for SMCF

Table 3-783: 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

Specifies the number of alerts present A value of 0 means that no alerts are present.

ALT_ADDR_CFG**Type**

int

Default value

0x0

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

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

Specifies the number of data values(DATA_PER_MON_CFG+1) generated from each monitor.

DEF_CFG_IRQ_MASK**Type**

int

Default value

0x1

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

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

Specifies the default alternate read address. This sets the default value for MGI_RADDR0/MGI_RADDR1.

DEF_WADDR_CFG**Type**

int

Default value

0x0

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

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

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.

FAKE_SENSOR_MAX_LIMIT**Type**

int

Default value

0xffff

Maximum value limit for the fake sensor/monitor.

FAKE_SENSOR_MIN_LIMIT**Type**

int

Default value

0x0

Minimum value limit for the fake sensor/monitor.

GRP_ID_CFG**Type**

int

Default value

0x0

Specifies a unique identifier for an MGI.

MGI_AIDR_RESET_VALUE**Type**

int

Default value

0x10

Reset value for MGI_AIDR register.

MGI_IIDR_RESET_VALUE**Type**

int

Default value

0x8d00043b

Reset value for MGI_IIDR register.

MLI_QUANTITY**Type**

int

Default value

0x1

Specifies the number of MLI.

MODE_LEN_CFG**Type**

int

Default value

0x1f

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

Specifies the number of MGI_MODE_REQ/STAT pairs. A value of 0 means that no MGI_MODE_REQ/STAT pairs are present.

MON_BASE_ADDRESS**Type**

int

Default value

0x0

AMU base address to which an offset is applied. It is used for configuration writes and data read bursts.

MON_DATA_WIDTH_CFG**Type**

int

Default value

0x1f

Specifies the bit width(MON_DATA_WIDTH_CFG+1) of each monitor data value.

MON_DISCON_CFG**Type**

int

Default value

0x0

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

Specifies the number of monitors(MON_NUM_CFG+1) in an MGI.

PACKED_CFG**Type**

int

Default value

0x0

Specifies if monitor data is packed 0: Data is not packed 1: Data is packed.

PER_TIMER_CFG**Type**

int

Default value

0x0

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

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

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

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

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

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

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

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

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.7.60 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 3-784: SMMUv3TestEngine Iris instances

InstanceName	ComponentName
SMMUv3TestEngine	SMMUv3TestEngine
SMMUv3TestEngine.register_file[0]	PVBusSlave

This model has the following MTI trace components:

Table 3-785: SMMUv3TestEngine MTI instances

InstanceName	ComponentName
SMMUv3TestEngine	SMMUv3TestEngine
SMMUv3TestEngine.register_file[0]	PVBusSlave

SMMUv3TestEngine contains the following CADI targets:

- SMMUv3TestEngine

Ports for SMMUv3TestEngine

Table 3-786: 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

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

The maximum number of in-flight transactions allowed.

output_attribute_transform

Type

string

Default value

""

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

The seed to use for initialising the random number generators.

3.7.61 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 3-787: STLBusGasket Iris instances

InstanceName	ComponentName
STLBusGasket	STLBusGasket
STLBusGasket.busmapper	PVBusMapper

This model has the following MTI trace components:

Table 3-788: STLBusGasket MTI instances

InstanceName	ComponentName
STLBusGasket.busmapper	PVBusMapper

STLBusGasket contains the following CADI targets:

- STLBusGasket

Ports for STLBusGasket

Table 3-789: Ports

Name	Protocol	Type	Description
pvbus_in	PVBus	Slave	-
pvbus_out	PVBus	Master	-

Parameters for STLBusGasket

function

Type

int

Default value

0x0

Function: 0-none, 1-STL value-forcing.

reg_base

Type
int

Default value
0xe001e820

Base Address of STL control regs (ADDR,VAL at offsets 0,4).

verbose

Type
int

Default value
0x0

Verbosity : 0-none, 1-some.

3.7.62 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 3-790: 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.

3.7.63 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 3-791: SoC_SOR Iris instances

InstanceName	ComponentName
SoC_SOR	SoC_SOR
SoC_SOR.PVBusMapper	PVBusMapper
SoC_SOR.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-792: SoC_SOR MTI instances

InstanceName	ComponentName
SoC_SOR.PVBusMapper	PVBusMapper
SoC_SOR.PVBusSlave	PVBusSlave

SoC_SOR contains the following CADI targets:

- SoC_SOR

Ports for SoC_SOR

Table 3-793: 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

Diagnostics.

gpr0

Type

int

Default value

0x0

General Purpose Register 0.

gpr1
Type
int
Default value
0x0

General Purpose Register 1.

3.7.64 SystemIdUnit

System ID Unit. This model is written in C++.

Iris and MTI instances for SystemIdUnit

This model has the following Iris instances:

Table 3-794: SystemIdUnit Iris instances

InstanceName	ComponentName
SystemIdUnit	SystemIdUnit
SystemIdUnit.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-795: SystemIdUnit MTI instances

InstanceName	ComponentName
SystemIdUnit.PVBusSlave	PVBusSlave

SystemIdUnit contains the following CADI targets:

- SystemIdUnit

Ports for SystemIdUnit

Table 3-796: Ports

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	-

Parameters for SystemIdUnit

chip_id
Type
int
Default value
0x0

the ID for the node/chip when there are multiple SoCs.

chiplet_type

Type

bool

Default value

0x0

Specifies Compute Chiplet (CC) or Specialization Chiplet (SC). 0 - CC, 1 - SC.

multi_chip_mode

Type

bool

Default value

0x0

Multi chip mode?.

platform_type

Type

int

Default value

0x0

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

the ID for the SoC that integrates the subsystem.

system_cfg

Type

int

Default value

0x0

the ID for the subsystem configuration.

system_id**Type**

int

Default value

0x0

the version ID for the subsystem.

3.7.65 System_Time_Synchronization_Control_cpp

System Time Synchronization Control. This model is written in C++.

Iris and MTI instances for System_Time_Synchronization_Control_cpp

This model has the following Iris instances:

Table 3-797: System_Time_Synchronization_Control_cpp Iris instances

InstanceName	ComponentName
System_Time_Synchronization_Control_cpp	System_Time_Synchronization_Control
System_Time_Synchronization_Control_cpp.PVBusMapper	PVBusMapper
System_Time_Synchronization_Control_cpp.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-798: System_Time_Synchronization_Control_cpp MTI instances

InstanceName	ComponentName
System_Time_Synchronization_Control_cpp.PVBusMapper	PVBusMapper
System_Time_Synchronization_Control_cpp.PVBusSlave	PVBusSlave

System_Time_Synchronization_Control_cpp contains the following CADI targets:

- System_Time_Synchronization_Control

Ports for System_Time_Synchronization_Control_cpp

Table 3-799: Ports

Name	Protocol	Type	Description
memmap_counter_m	PVBus	Master	-
pri_signal_s	Signal	Slave	-
pdbus_s	PVBus	Slave	-
sec_signal_m	Signal	Master	-
tupdate_pdbus_m	PVBus	Master	-
tupdate_pdbus_s	PVBus	Slave	-

Parameters for System_Time_Synchronization_Control_cpp

diagnostics

Type
int
Default value
0x0

Diagnostics.

is_manager_chip

Type
bool
Default value
0x0

Set to true if the Time Synchronization module is present in the primary chip or else false.

3.7.66 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 3-800: TRNG Iris instances

InstanceName	ComponentName
TRNG	TRNG
TRNG.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-801: TRNG MTI instances

InstanceName	ComponentName
TRNG.PVBusSlave	PVBusSlave

TRNG contains the following CADI targets:

- TRNG

Ports for TRNG

Table 3-802: 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

Diagnostics.

3.7.67 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++.

Iris and MTI instances for TelnetTerminal

This model has the following Iris instances:

Table 3-803: TelnetTerminal Iris instances

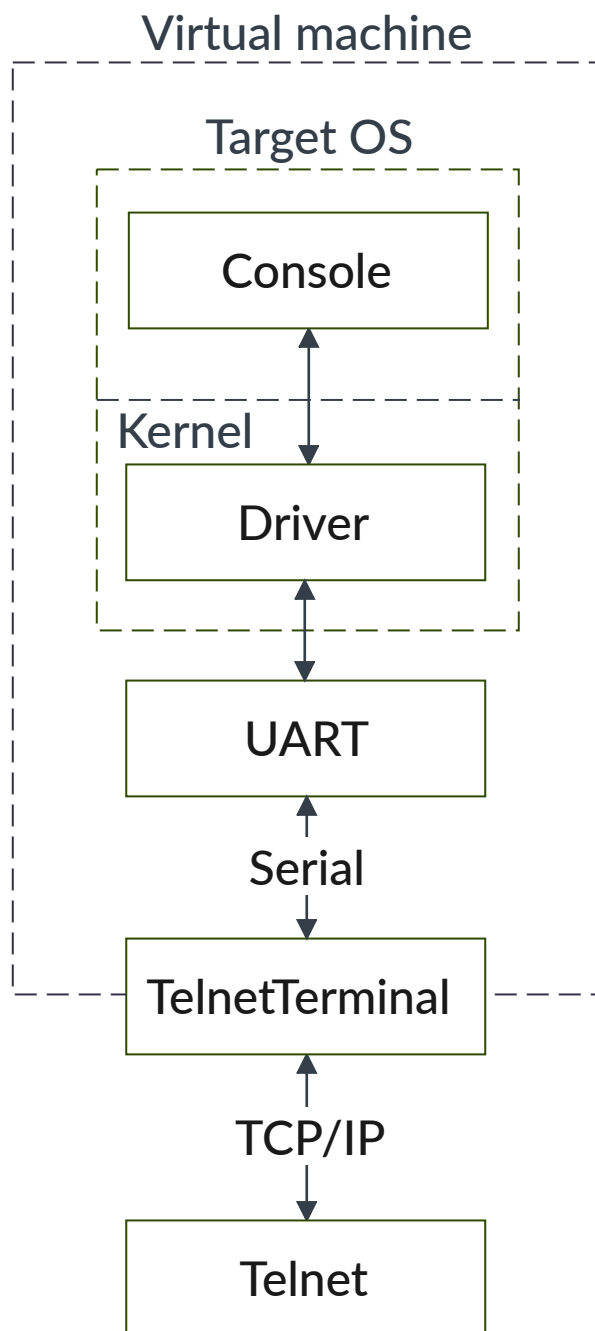
InstanceName	ComponentName
TelnetTerminal	TelnetTerminal

TelnetTerminal contains the following CADI targets:

- TelnetTerminal

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 [3.10.65 PL011_Uart](#) on page 4486. 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.

Microsoft Windows 10 disables the Telnet client by default. Follow these steps to enable it:



Note

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.



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"
```

Ports for TelnetTerminal

Table 3-804: 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"

Terminal initialisation mode.

quiet

Type

bool

Default value

0x0

Avoid output on stdout/stderr.

start_port

Type
int

Default value
0x1388

Telnet TCP Port Number.

start_telnet

Type
bool

Default value
0x1

Start telnet if nothing connected.

terminal_command

Type
string

Default value
""

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).

3.7.68 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 3-805: Temperature Iris instances

InstanceName	ComponentName
Temperature	Temperature

Temperature contains the following CADI targets:

- Temperature

Ports for Temperature

Table 3-806: 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

Temperature Coefficient.

diagnostics

Type

int

Default value

0x0

Diagnostics.

num_cores

Type

int

Default value

0x4

Number of cores.

3.7.69 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 3-807: UnusedPrimeCell Iris instances

InstanceName	ComponentName
UnusedPrimeCell	UnusedPrimeCell
UnusedPrimeCell.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-808: UnusedPrimeCell MTI instances

InstanceName	ComponentName
UnusedPrimeCell.busslave	PVBusSlave

UnusedPrimeCell contains the following CADI targets:

- UnusedPrimeCell

Ports for UnusedPrimeCell

Table 3-809: Ports

Name	Protocol	Type	Description
pvbuss	PVBus	Slave	Bus slave interface.

3.7.70 VirtioBlockDevice

virtio block device. This model is written in C++.

Iris and MTI instances for VirtioBlockDevice

This model has the following Iris instances:

Table 3-810: VirtioBlockDevice Iris instances

InstanceName	ComponentName
VirtioBlockDevice	VirtioBlockDevice
VirtioBlockDevice.register_slave	PVBusSlave
VirtioBlockDevice.virtio_master	PVBusMaster

This model has the following MTI trace components:

Table 3-811: VirtioBlockDevice MTI instances

InstanceName	ComponentName
VirtioBlockDevice	VirtioBlockDevice
VirtioBlockDevice.register_slave	PVBusSlave
VirtioBlockDevice.virtio_master	PVBusMaster

VirtioBlockDevice contains the following CADI targets:

- VirtioBlockDevice

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.

Ports for VirtioBlockDevice

Table 3-812: 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

""

image file path.

`quiet`

Type

bool

Default value

0x0

Don't print warnings on malformed commands/descriptors.

`read_only`

Type

bool

Default value

0x0

Only allow device to be read.

secure_accesses

Type
bool

Default value
0x0

Make device generate transactions with NS=0.

transaction_attributes

Type
int

Default value
0x0

Transaction attributes used by device. 0x0 - inner-shared real access. 0x1 - outer-shared real access. 0x2 - outer-shared debug access.

3.7.71 VirtioBlockDeviceMMIO

Virtio v1.0 Block device over MMIO transport. This model is written in C++.

Iris and MTI instances for VirtioBlockDeviceMMIO

This model has the following Iris instances:

Table 3-813: VirtioBlockDeviceMMIO Iris instances

InstanceName	ComponentName
VirtioBlockDeviceMMIO	VirtioBlockMMIO
VirtioBlockDeviceMMIO.dma_master	PVBusMaster

This model has the following MTI trace components:

Table 3-814: VirtioBlockDeviceMMIO MTI instances

InstanceName	ComponentName
VirtioBlockDeviceMMIO.dma_master	PVBusMaster

VirtioBlockDeviceMMIO contains the following CADI targets:

- VirtioBlockMMIO

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.

Ports for VirtioBlockDeviceMMIO

Table 3-815: 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

Enable or disable device. If disabled, device can be accessed, but will not be activated.

image_path

Type

string

Default value

""

Image file path.

quiet

Type

bool

Default value

0x0

Don't print info or warnings (e.g. on malformed commands/descriptors).

read_only

Type

bool

Default value

0x0

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

Make device generate transactions with NS=0.

transaction_attributes

Type

int

Default value

0x0

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"

Choose legacy or modern virtio transport, if not specified, modern transport is used.

3.7.72 VirtioNetMMIO

Virtio net device over MMIO transport. This model is written in C++.

Iris and MTI instances for VirtioNetMMIO

This model has the following Iris instances:

Table 3-816: VirtioNetMMIO Iris instances

InstanceName	ComponentName
VirtioNetMMIO	VirtioNetMMIO
VirtioNetMMIO.dma_master	PVBusMaster
VirtioNetMMIO.hostbridge	HostBridge

This model has the following MTI trace components:

Table 3-817: VirtioNetMMIO MTI instances

InstanceName	ComponentName
VirtioNetMMIO.dma_master	PVBusMaster

VirtioNetMMIO contains the following CADI targets:

- HostBridge
- VirtioNetMMIO

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 Microsoft Windows](#) and [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 Microsoft Windows](#)
- [Configuring the networking environment for Linux](#)

Ports for VirtioNetMMIO

Table 3-818: 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

""

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

Enable or disable device. If disabled, device can be accessed, but will not be activated.

hostbridge.interfaceName**Type**

string

Default value

""

Host Interface.

hostbridge.userNetOptions**Type**

string

Default value

""

Control options for UserNet TCP/IP (for internal use only, please do not use).

hostbridge.userNetPorts**Type**

string

Default value

""

Listening ports to expose in user-mode networking.

hostbridge.userNetSubnet**Type**

string

Default value

"172.20.51.0/24"

Virtual subnet for user-mode networking.

hostbridge.userNetworking**Type**

bool

Default value

0x1

Enable user-mode networking.

mac_address**Type**

string

Default value

""

Device MAC address, if not specified, a random MAC address is generated.

offload**Type**

string

Default value

"gso, gro"

Offload TCP/UDP segmentation/receiving operations to host.

secure_accesses**Type**

bool

Default value

0x0

Make device generate transactions with NS=0.

transport**Type**

string

Default value

"modern"

Choose legacy or modern virtio transport, if not specified, modern transport is used.

3.7.73 VirtioP9Device

virtio P9 server. This model is written in C++.

Iris and MTI instances for VirtioP9Device

This model has the following Iris instances:

Table 3-819: VirtioP9Device Iris instances

InstanceName	ComponentName
VirtioP9Device	VirtioP9Device
VirtioP9Device.mmio_slave	PVBUSSlave
VirtioP9Device.virtio_master	PVBUSMaster

This model has the following MTI trace components:

Table 3-820: VirtioP9Device MTI instances

InstanceName	ComponentName
VirtioP9Device	VirtioP9Device
VirtioP9Device.mmio_slave	PVBUSSlave
VirtioP9Device.virtio_master	PVBUSMaster

VirtioP9Device contains the following CADI targets:

- [VirtioP9Device](#)

About VirtioP9Device

This component implements a subset of the Plan 9 file protocol over a virtio transport. It enables accessing 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.

It 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, Tsymlink.
- On Windows hosts, it ignores Unix permissions when writing files.
- On Windows hosts, it performs a simple mapping from Windows to Unix permissions when reading.
- On Windows hosts, symbolic links appear as regular files.
- On Windows hosts, 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.virtiop9device.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>;
};
```

Ports for VirtioP9Device

Table 3-821: 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"

mount tag.

quiet
Type
bool
Default value
0x0

Don't print warnings on malformed commands/descriptors.

root_path
Type
string
Default value
""

root directory path.

secure_accesses
Type
bool
Default value
0x0

Make device generate transactions with NS=0.

3.7.74 VirtioPCIBlockDevice

virtio block device with PCI transport. This model is written in C++.

Iris and MTI instances for VirtioPCIBlockDevice

This model has the following Iris instances:

Table 3-822: VirtioPCIBlockDevice Iris instances

InstanceName	ComponentName
VirtioPCIBlockDevice	VirtioPCIBlockDevice
VirtioPCIBlockDevice.register_slave	PVBusSlave
VirtioPCIBlockDevice.virtio_master	PVBusMaster

This model has the following MTI trace components:

Table 3-823: VirtioPCIBlockDevice MTI instances

InstanceName	ComponentName
VirtioPCIBlockDevice	VirtioPCIBlockDevice
VirtioPCIBlockDevice.register_slave	PVBusSlave
VirtioPCIBlockDevice.virtio_master	PVBusMaster

VirtioPCIBlockDevice contains the following CADI targets:

- VirtioPCIBlockDevice

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

Ports for VirtioPCIBlockDevice

Table 3-824: 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

""

image file path.

quiet**Type**

bool

Default value

0x0

Don't print warnings on malformed commands/descriptors.

read_only**Type**

bool

Default value

0x0

Only allow device to be read.

secure_accesses**Type**

bool

Default value

0x0

Make device generate transactions with NS=0.

transaction_attributes**Type**

int

Default value

0x0

Transaction attributes used by device. 0x0 - inner-shared real access. 0x1 - outer-shared real access. 0x2 - outer-shared debug access.

3.7.75 VirtioRNG

virtio rng - Entropy device. This model is written in C++.

Iris and MTI instances for VirtioRNG

This model has the following Iris instances:

Table 3-825: VirtioRNG Iris instances

InstanceName	ComponentName
VirtioRNG	VirtioEntropyMMIO
VirtioRNG.dma_master	PVBusMaster

This model has the following MTI trace components:

Table 3-826: VirtioRNG MTI instances

InstanceName	ComponentName
VirtioRNG.dma_master	PVBusMaster

VirtioRNG contains the following CADI targets:

- [VirtioEntropyMMIO](#)

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.pybus_m_range[0x001C190000..0x001C19ffff] => virtio_rng.pybus;
    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
```

Ports for VirtioRNG

Table 3-827: Ports

Name	Protocol	Type	Description
intr	Signal	Master	Virtio device sets interrupt to signal completion.
pvbust	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

Prints debug information: 0 = disabled; 1 = generated seed and device; 4 = generated seed, device and generated numbers.

enabled**Type**

bool

Default value

0x0

Enable or disable device. If disabled, device can be accessed, but will not be activated.

generator**Type**

int

Default value

0x0

User-defined generator: 0 = xorshiftstar; 1 = rand48; 2 = mersenne.

secure_accesses**Type**

bool

Default value

0x0

Make device generate transactions with NS=0.

seed**Type**

int

Default value

0x0

User-defined seed: 0 = uses a random seed; > 0 = user-defined fixed seed value.

transport**Type**

string

Default value

"modern"

Choose legacy or modern virtio transport, if not specified, modern transport is used.

3.7.76 VirtualEthernetCrossover

Ethernet Crossover Cable. This model is written in LISA+.

Iris and MTI instances for VirtualEthernetCrossover

This model has the following Iris instances:

Table 3-828: VirtualEthernetCrossover Iris instances

InstanceName	ComponentName
VirtualEthernetCrossover	VirtualEthernetCrossover

This model has the following MTI trace components:

Table 3-829: VirtualEthernetCrossover MTI instances

InstanceName	ComponentName
VirtualEthernetCrossover	VirtualEthernetCrossover

VirtualEthernetCrossover contains the following CADI targets:

- VirtualEthernetCrossover

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.

Ports for VirtualEthernetCrossover

Table 3-830: 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.

3.7.77 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 3-831: VirtualEthernetHub3 Iris instances

InstanceName	ComponentName
VirtualEthernetHub3	VirtualEthernetHub3

VirtualEthernetHub3 contains the following CADI targets:

- VirtualEthernetHub3

Ports for VirtualEthernetHub3

Table 3-832: 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.

3.7.78 VisEventRecorder

Event recorder component for visualisation component. It allows you to play back and record interactive GUI sessions. This model is written in LISA+.

Iris and MTI instances for VisEventRecorder

This model has the following Iris instances:

Table 3-833: 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 3-834: VisEventRecorder MTI instances

InstanceName	ComponentName
VisEventRecorder.playbackDivider	ClockDivider
VisEventRecorder.recordingDivider	ClockDivider

VisEventRecorder contains the following CADI targets:

- ClockDivider
- ClockTimerThread
- ClockTimerThread64
- SchedulerThread
- SchedulerThreadEvent
- VisEventRecorder

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()`).

Ports for VisEventRecorder

Table 3-835: 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

check instruction count in recording file against actual instruction count during playback.

playbackDivider.div

Type

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

playbackFileName

Type

string

Default value

""

playback filename (empty string disables playback).

recordingDivider.div

Type

int

Default value

0x1

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

recordingFileName

Type

string

Default value

""

recording filename (empty string disables recording).

recordingTimeBase

Type

int

Default value

0x5f5e100

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

enable verbose messages (1=normal, 2=even more).

3.7.79 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 3-836: Visualisation_sdl2 Iris instances

InstanceName	ComponentName
Visualisation_sdl2	Visualisation_sdl2
Visualisation_sdl2.VisEventRecorder_cpp	VisEventRecorder
Visualisation_sdl2.VisEventRecorder_cpp.ClockDivider	ClockDivider

InstanceName	ComponentName
Visualisation_sdl2.VisEventRecorder_cpp.ClockTimerThread	ClockTimerThread
Visualisation_sdl2.VisEventRecorder_cpp.ClockTimerThread.ClockTimerThread64	ClockTimerThread64
Visualisation_sdl2.VisEventRecorder_cpp.ClockTimerThread.ClockTimerThread64.SchedulerThread	SchedulerThread
Visualisation_sdl2.VisEventRecorder_cpp.ClockTimerThread.ClockTimerThread64.SchedulerThreadEvent	SchedulerThreadEvent

This model has the following MTI trace components:

Table 3-837: Visualisation_sdl2 MTI instances

InstanceName	ComponentName
Visualisation_sdl2.VisEventRecorder_cpp.ClockDivider	ClockDivider

Visualisation_sdl2 contains the following CADI targets:

- Visualisation_sdl2

Ports for Visualisation_sdl2

Table 3-838: Ports

Name	Protocol	Type	Description
c0_core_freq[16]	ValueState	Slave	-
c10_core_freq[16]	ValueState	Slave	-
c11_core_freq[16]	ValueState	Slave	-
c12_core_freq[16]	ValueState	Slave	-
c13_core_freq[16]	ValueState	Slave	-
c14_core_freq[16]	ValueState	Slave	-
c15_core_freq[16]	ValueState	Slave	-
c1_core_freq[16]	ValueState	Slave	-
c2_core_freq[16]	ValueState	Slave	-
c3_core_freq[16]	ValueState	Slave	-
c4_core_freq[16]	ValueState	Slave	-
c5_core_freq[16]	ValueState	Slave	-
c6_core_freq[16]	ValueState	Slave	-
c7_core_freq[16]	ValueState	Slave	-
c8_core_freq[16]	ValueState	Slave	-
c9_core_freq[16]	ValueState	Slave	-
clock_50Hz	ClockSignal	Slave	-
cluster0_ticks[16]	InstructionCount	Slave	-
cluster10_ticks[16]	InstructionCount	Slave	-
cluster11_ticks[16]	InstructionCount	Slave	-
cluster12_ticks[16]	InstructionCount	Slave	-
cluster13_ticks[16]	InstructionCount	Slave	-

Name	Protocol	Type	Description
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

""

Cluster0 name.

cluster10_name

Type

string

Default value

""

Cluster10 name.

cluster11_name

Type

string

Default value

""

Cluster11 name.

cluster12_name

Type

string

Default value

""

Cluster12 name.

cluster13_name

Type

string

Default value

""

Cluster13 name.

cluster14_name

Type

string

Default value

""

Cluster14 name.

cluster15_name

Type

string

Default value

""

Cluster15 name.

cluster1_name**Type**

string

Default value

""

Cluster1 name.

cluster2_name**Type**

string

Default value

""

Cluster2 name.

cluster3_name**Type**

string

Default value

""

Cluster3 name.

cluster4_name**Type**

string

Default value

""

Cluster4 name.

cluster5_name**Type**

string

Default value

""

Cluster5 name.

cluster6_name**Type**

string

Default value

""

Cluster6 name.

cluster7_name**Type**

string

Default value

""

Cluster7 name.

cluster8_name**Type**

string

Default value

""

Cluster8 name.

cluster9_name**Type**

string

Default value

""

Cluster9 name.

css_spec**Type**

string

Default value

"Columbus mid"

Platform specification displayed in window title.

diagnostics**Type**

int

Default value

0x0

Diagnostics.

disable_visualisation**Type**

bool

Default value

0x0

Enable/disable visualisation.

display_object**Type**

int

Default value

0x0

Display objects: LCD only (1), status only (2), or both (0).

display_poreset_button**Type**

bool

Default value

0x0

Display power-on reset button.

idler.delay_ms**Type**

int

Default value

0x32

GUI update period in ms.

idler.has_gui**Type**

bool

Default value

0x1

GUI is enabled.

is_heterogeneous_cluster**Type**

bool

Default value

0x0

Is Heterogeneous cluster.

lcd_height_param**Type**

int

Default value

0x258

LCD Height.

lcd_width_param**Type**

int

Default value

0x320

LCD Width.

mcp_name**Type**

string

Default value

"MCP: Cortex-M7"

MCP name.

num_cps**Type**

int

Default value

0x1

Number of Control Processors.

per_core_clock**Type**

bool

Default value

0x0

Per-core clock connection.

platform_name**Type**

string

Default value

"Application Processors"

Platform Name.

rate_limit-enable**Type**

bool

Default value

0x0

Rate limit simulation.

recorder.checkInstructionCount**Type**

bool

Default value

0x1

check instruction count in recording file against actual instruction count during playback.

recorder.playbackDivider.div**Type**

int

Default value

0x1

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

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

recorder.playbackFileName**Type**

string

Default value

""

playback filename (empty std::string disables playback).

recorder.recordingDivider.div**Type**

int

Default value

0x1

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

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

recorder.recordingFileName**Type**

string

Default value

""

recording filename (empty std::string disables recording).

recorder.recordingTimeBase**Type**

int

Default value

0x5f5e100

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

enable verbose messages (1=normal, 2=even more).

scp_name**Type**

string

Default value

"SCP: Cortex-M7"

SCP name.

shutdown_pixel_enable**Type**

bool

Default value

0x0

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

Shutdown pixel target RGB value 0xRRGGBB.

shutdown_pixel_x**Type**

int

Default value

0x0

Shutdown pixel X co-ordinate (0 is left).

shutdown_pixel_y**Type**

int

Default value

0x0

Shutdown pixel Y co-ordinate (0 is top).

trap_key**Type**

int

Default value

0x4a

Trap key that works with left Ctrl to toggle mouse display.

window_title**Type**

string

Default value

"Fast Models - %cpu%"

Window title(%cpu% will be replaced by css_spec).

3.7.80 WarningMemory

Memory that prints warnings, and RAZ/WIs or aborts. This model is written in C++.

Ports for WarningMemory

Table 3-839: Ports

Name	Protocol	Type	Description
pvbust	PVBus	Slave	Bus slave interface

Parameters for WarningMemory

`abort_on_reads`

Type

bool

Default value

0x0

Generate Abort on reads.

`abort_on_writes`

Type

bool

Default value

0x0

Generate Abort on writes.

`read_data`

Type

int

Default value

0x0

Data to return on reads, if not aborting.

`warn_on_reads`

Type

bool

Default value

0x1

Generate Warn on reads.

warn_on_writes

Type
bool

Default value
0x1

Generate Warn on writes.

warning

Type
string

Default value
"Invalid access"

Warning string.

3.7.81 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];
...
cluster4.cti[3] => v8ect.cti[12];
cluster4.cti[3] => v8ect.cti[13];
cluster4.cti[3] => v8ect.cti[14];
cluster4.cti[3] => v8ect.cti[15];
```

Ports for v8EmbeddedCrossTrigger_Matrix

Table 3-840: Ports

Name	Protocol	Type	Description
cti[36]	v8EmbeddedCrossTrigger_controlprotocol	Slave	-

Parameters for v8EmbeddedCrossTrigger_Matrix

has_CTIAUTHSTATUS

Type

bool

Default value

0x1

Has the optional CTIAUTHSTATUS register.

has_CTIDEVID_INOUT

Type

bool

Default value

0x1

Has the option of input gate in cross trigger matrix.

number-of-channels

Type

int

Default value

0x4

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 3-841: Ports

Name	Protocol	Type	Description
async_callback	AsyncSignalCallback	Master	This port emits a call to signal() on the simulation thread asynchronously after async_control.signal() has been called.
async_control	AsyncSignalControl	Slave	Non-simulation threads call signal() on this port in order to schedule an event: a call to async_callback.signal() 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 3-842: SchedulerInterface Iris instances

InstanceName	ComponentName
SchedulerInterface	SchedulerInterface

SchedulerInterface contains the following CADI targets:

- SchedulerInterface

Ports for SchedulerInterface

Table 3-843: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock frequency for waitTicks() function.
control	SchedulerInterfaceControl	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 3-844: SchedulerThread Iris instances

InstanceName	ComponentName
SchedulerThread	SchedulerThread

SchedulerThread contains the following CADI targets:

- SchedulerThread

Ports for SchedulerThread

Table 3-845: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock frequency for waitTicks() function.
control	SchedulerThreadControl	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 3-846: SchedulerThreadEvent Iris instances

InstanceName	ComponentName
SchedulerThreadEvent	SchedulerThreadEvent

SchedulerThreadEvent contains the following CADI targets:

- SchedulerThreadEvent

Ports for SchedulerThreadEvent

Table 3-847: Ports

Name	Protocol	Type	Description
control	SchedulerThreadEventControl	Slave	SchedulerThreadEvent control. Masters use this to: - wait for this event - notify waiters that the event happened

3.8.5 SchedulerThreadEvent_cpp

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on. This model is written in C++.

Iris and MTI instances for SchedulerThreadEvent_cpp

This model has the following Iris instances:

Table 3-848: SchedulerThreadEvent_cpp Iris instances

InstanceName	ComponentName
SchedulerThreadEvent_cpp	SchedulerThreadEvent

SchedulerThreadEvent_cpp contains the following CADI targets:

- SchedulerThreadEvent

Ports for SchedulerThreadEvent_cpp

Table 3-849: Ports

Name	Protocol	Type	Description
control	SchedulerThreadEventControl	Slave	-

3.8.6 SchedulerThread_cpp

A SchedulerThread instance represents a co-routine thread in the simulation. This model is written in C++.

Iris and MTI instances for SchedulerThread_cpp

This model has the following Iris instances:

Table 3-850: SchedulerThread_cpp Iris instances

InstanceName	ComponentName
SchedulerThread_cpp	SchedulerThread

SchedulerThread_cpp contains the following CADI targets:

- SchedulerThread

Ports for SchedulerThread_cpp

Table 3-851: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-
control	SchedulerThreadControl	Slave	-

3.9 Signals components

This section describes the Signals components.

3.9.1 AndGate

And Gate. This model is written in LISA+.

Iris and MTI instances for AndGate

This model has the following Iris instances:

Table 3-852: AndGate Iris instances

InstanceName	ComponentName
AndGate	AndGate

AndGate contains the following CADI targets:

- AndGate

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.

Ports for AndGate

Table 3-853: Ports

Name	Protocol	Type	Description
input[2]	Signal	Slave	2 input signals to be AND'ed.
output	Signal	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 3-854: FrequencyProbe Iris instances

InstanceName	ComponentName
FrequencyProbe	FrequencyProbe

FrequencyProbe contains the following CADI targets:

- FrequencyProbe

Ports for FrequencyProbe

Table 3-855: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-
freq_changed	ValueState	Master	-

Parameters for FrequencyProbe

diagnostics

Type

int

Default value

0x0

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 3-856: 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 3-857: LabellerMasterIdExtendedIdUserFlag MTI instances

InstanceName	ComponentName
LabellerMasterIdExtendedIdUserFlag.pvbuslogger	PVBusLogger
LabellerMasterIdExtendedIdUserFlag.pvbuslogger.mapper	PVBusMapper
LabellerMasterIdExtendedIdUserFlag.pvbusmodifier	PVBusMapper

LabellerMasterIdExtendedIdUserFlag contains the following CADI targets:

- LabellerMasterIdExtendedIdUserFlag
- PVBusLogger

Ports for LabellerMasterIdExtendedIdUserFlag**Table 3-858: 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

ExtendedID value to be applied to transactions.

extendedid_mask

Type

int

Default value

0x0

Mask used to determine which bits of extendedid parameter to be set in the transactions ExtendedID attribute. 0xFFFFFFFF will overwrite all the incoming ExtendedID bits with the value of the extendedid parameter, 0x0 will overwrite none.

masterid

Type

int

Default value

0x0

MasterID value to be applied to transactions.

masterid_mask

Type

int

Default value

0x0

Mask used to determine which bits of masterid parameter to be set in the transactions MasterID attribute. 0xFFFF 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

Enable tracing of debug transactions.

pvbuslogger.trace_snoops

Type
bool

Default value
0x0

Enable tracing of ACE snoop requests.

userflags

Type
int

Default value
0x0

UserFlags value to be applied to transactions.

userflags_mask

Type
int

Default value
0x0

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 3-859: 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 3-860: LabellerUserSignals MTI instances

InstanceName	ComponentName
LabellerUserSignals.pvbuslogger	PVBusLogger
LabellerUserSignals.pvbuslogger.mapper	PVBusMapper
LabellerUserSignals.pvbusmodifier	PVBusMapper

LabellerUserSignals contains the following CADI targets:

- LabellerUserSignals
- PVBusLogger

Ports for LabellerUserSignals

Table 3-861: 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

Enable tracing of debug transactions.

pvbuslogger.trace_snoops

Type

bool

Default value

0x0

Enable tracing of ACE snoop requests.

user

Type

int

Default value

0x0

User signal to be applied to transactions.

3.9.5 OrGate

Or Gate. This model is written in LISA+.

Iris and MTI instances for OrGate

This model has the following Iris instances:

Table 3-862: OrGate Iris instances

InstanceName	ComponentName
OrGate	OrGate

OrGate contains the following CADI targets:

- OrGate

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.

Ports for OrGate

Table 3-863: 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+.

Iris and MTI instances for SGSignalBuffer

This model has the following Iris instances:

Table 3-864: SGSignalBuffer Iris instances

InstanceName	ComponentName
SGSignalBuffer	SGSignalBuffer

SGSignalBuffer contains the following CADI targets:

- SGSignalBuffer



Variants of this component also exist with multiple input and output ports.

Ports for SGSignalBuffer

Table 3-865: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock rate to release buffered signals.
in	Signal	Slave	Signal in.
out	Signal	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 3-866: SignalDriver Iris instances

InstanceName	ComponentName
SignalDriver	SignalDriver
SignalDriver.pvbuslave	PVBusSlave

This model has the following MTI trace components:

Table 3-867: SignalDriver MTI instances

InstanceName	ComponentName
SignalDriver	SignalDriver
SignalDriver.pvbuslave	PVBusSlave

SignalDriver contains the following CADI targets:

- SignalDriver

Ports for SignalDriver

Table 3-868: Ports

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	-
signal_out	Signal	Master	-

Parameters for SignalDriver

param_input

Type

bool

Default value

0x0

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 3-869: SignalInverter Iris instances

InstanceName	ComponentName
SignalInverter	SignalInverter

SignalInverter contains the following CADI targets:

- SignalInverter

Ports for SignalInverter**Table 3-870: 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 3-871: SignalLogger Iris instances

InstanceName	ComponentName
SignalLogger	SignalLogger

This model has the following MTI trace components:

Table 3-872: SignalLogger MTI instances

InstanceName	ComponentName
SignalLogger	SignalLogger

SignalLogger contains the following CADI targets:

- SignalLogger

Ports for SignalLogger

Table 3-873: 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

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 3-874: Value64Logger Iris instances

InstanceName	ComponentName
Value64Logger	Value64Logger

This model has the following MTI trace components:

Table 3-875: Value64Logger MTI instances

InstanceName	ComponentName
Value64Logger	Value64Logger

Value64Logger contains the following CADI targets:

- Value64Logger

Ports for Value64Logger

Table 3-876: 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 3-877: ValueLogger Iris instances

InstanceName	ComponentName
ValueLogger	ValueLogger

This model has the following MTI trace components:

Table 3-878: ValueLogger MTI instances

InstanceName	ComponentName
ValueLogger	ValueLogger

ValueLogger contains the following CADI targets:

- ValueLogger

Ports for ValueLogger

Table 3-879: 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 3-880: WideAndGate Iris instances

InstanceName	ComponentName
WideAndGate	WideAndGate

WideAndGate contains the following CADI targets:

- WideAndGate

Ports for WideAndGate

Table 3-881: Ports

Name	Protocol	Type	Description
input[8]	Signal	Slave	-
output	Signal	Master	-

Parameters for WideAndGate

diagnostics

Type

int

Default value

0x0

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 3-882: WideOrGate Iris instances

InstanceName	ComponentName
WideOrGate	WideOrGate

WideOrGate contains the following CADI targets:

- WideOrGate

Ports for WideOrGate

Table 3-883: Ports

Name	Protocol	Type	Description
input[8]	Signal	Slave	-
output	Signal	Master	-

Parameters for WideOrGate

diagnostics

Type
int
Default value
0x0

Diagnostics.

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 3-884: 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 23.

Iris and MTI instances for AHCI_SATA

This model has the following Iris instances:

Table 3-885: 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 3-886: AHCI_SATA MTI instances

InstanceName	ComponentName
AHCI_SATA	ExportTestAHCI_SATA
AHCI_SATA.ahci_master	PVBusMaster
AHCI_SATA.register_slave	PVBusSlave

AHCI_SATA contains the following CADI targets:

- AHCI_SATA

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.

Ports for AHCI_SATA

Table 3-887: 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"

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

""

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

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 3-888: 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 23.

Iris and MTI instances for AddressTranslationUnit

This model has the following Iris instances:

Table 3-889: AddressTranslationUnit Iris instances

InstanceName	ComponentName
AddressTranslationUnit	AddressTranslationUnit
AddressTranslationUnit.ATU_BusMapper	PVBusMapper
AddressTranslationUnit.apb	PVBusSlave

This model has the following MTI trace components:

Table 3-890: AddressTranslationUnit MTI instances

InstanceName	ComponentName
AddressTranslationUnit.ATU_BusMapper	PVBusMapper
AddressTranslationUnit.apb	PVBusSlave

AddressTranslationUnit contains the following CADI targets:

- AddressTranslationUnit

Ports for AddressTranslationUnit

Table 3-891: Ports

Name	Protocol	Type	Description
apb	PVBus	Slave	-
irq_out	Signal	Master	-
pvbus_m	PVBus	Master	-
pvbus_s	PVBus	Slave	-
reset_in	Signal	Slave	-

Parameters for AddressTranslationUnit

ATUNTR

Type

int

Default value

0x5

Number of translation regions (1=2, 2=4, 3=8, 4=16, and 5=32).

ATUPAW

Type

int

Default value

0x5

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

Selects the page size granularity in bytes (0xC=4096, 0xD=8192, and 0xE=16384) (Default=0xD).

diagnostics

Type

int

Default value

0x0

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 3-892: 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 23.

Iris and MTI instances for BP141_TZMA

This model has the following Iris instances:

Table 3-893: 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 3-894: BP141_TZMA MTI instances

InstanceName	ComponentName
BP141_TZMA.pvbusrange_0.pvbus_mapper	PVBusMapper
BP141_TZMA.tzswitch_0.pvbus_mapper	PVBusMapper

BP141_TZMA contains the following CADI targets:

- BP141_TZMA
- PVBusRange

- TZSwitch

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 3-895: 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

Ports for BP141_TZMA

Table 3-896: 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

Addressable range of device.

TZSECROMSIZE

Type
int

Default value
0x200

Default secure size.

TZSEGSIZE

Type
int

Default value
0x1000

Segment size.

pvbusrange_0.range

Type
int

Default value
0x0

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 3-897: 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 23.

Iris and MTI instances for BP147_TZPC

This model has the following Iris instances:

Table 3-898: BP147_TZPC Iris instances

InstanceName	ComponentName
BP147_TZPC	BP147_TZPC
BP147_TZPC.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-899: BP147_TZPC MTI instances

InstanceName	ComponentName
BP147_TZPC.busslave	PVBusSlave

BP147_TZPC contains the following CADI targets:

- BP147_TZPC

About BP147_TZPC

BP147_TZPC provides a software interface to the protection bits in a secure system in a TrustZone design.

Ports for BP147_TZPC

Table 3-900: 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 3-901: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for CCI400

This model has the following Iris instances:

Table 3-902: CCI400 Iris instances

InstanceName	ComponentName
CCI400	CCI400
CCI400.cciinterconnect	PVCache
CCI400.cciregisters	CCIRegisters
CCI400.cciregisters.clocktimer	ClockTimerThread

InstanceName	ComponentName
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 3-903: CCI400 MTI instances

InstanceName	ComponentName
CCI400	CCI400

CCI400 contains the following CADI targets:

- CCI400

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.

Ports for CCI400

Table 3-904: 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.

Name	Protocol	Type	Description
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.
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

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

barrierterminate

Type

int

Default value

0x7

For each downstream port, determine if barriers will be terminated at that port.

broadcastcachemain**Type**

int

Default value

0x0

For each downstream port a bit determines if broadcast cache maintenance operations are forwarded down that port.

bufferableoverride**Type**

int

Default value

0x0

For each downstream port, determine if all transactions will be forced to non-bufferable.

cache_state_modelled**Type**

bool

Default value

0x1

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

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

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"`

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 3-905: IP revisions support

Revision	Quality level
rOp0	Full support
rOp2	Full support
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.**Iris and MTI instances for CCI500**

This model has the following Iris instances:

Table 3-906: 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 3-907: CCI500 MTI instances

InstanceName	ComponentName
CCI500	CCI500
CCI500.pvbus_register_file_s[0]	PVBusSlave
CCI500.upstream[0]	PVBusSlave
CCI500.upstream[1]	PVBusSlave
CCI500.upstream[2]	PVBusSlave

InstanceName	ComponentName
CCI500.upstream[3]	PVBusSlave
CCI500.upstream[4]	PVBusSlave
CCI500.upstream[5]	PVBusSlave
CCI500.upstream[6]	PVBusSlave

CCI500 contains the following CADI targets:

- CCI500

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.

Ports for CCI500

Table 3-908: 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.

Name	Protocol	Type	Description
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.
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.
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 `pvbus_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 `pvbus_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_s[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 `pvbuss_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 `pvbuss_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

The bit-width of the address that the CCI can accept.

cache_state_modelled**Type**

bool

Default value

0x1

Model the cache state.

dbgen**Type**

bool

Default value

0x1

Invasive debug enable. If true, enables the counting of PMU events.

enable_logger**Type**

bool

Default value

0x0

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

Whether non-secure events are allowed to be counted in the performance monitor.

num_ace_lite_ports

Type

int

Default value

0x5

The bottom `num_ace_lite_ports` are ACE-Lite+DVM.

num_ace_ports

Type

int

Default value

0x2

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

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 3-909: 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 23.

Iris and MTI instances for CCI550

This model has the following Iris instances:

Table 3-910: 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

InstanceName	ComponentName
CCI550.upstream[4]	PVBusSlave
CCI550.upstream[5]	PVBusSlave
CCI550.upstream[6]	PVBusSlave

This model has the following MTI trace components:

Table 3-911: 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

CCI550 contains the following CADI targets:

- CCI550

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.

Ports for CCI550

Table 3-912: Ports

Name	Protocol	Type	Description
<code>acchannelensx[7]</code>	Value	Slave	The <code>acchannelensx[N]</code> pins are used to tell the interconnect if the upstream system will accept snoops and/or DVM messages.
<code>address_decoder</code>	CCI500_ AddressDecoderProtocol	Master	An address decoder can be attached to the <code>address_decoder</code> port to choose which <code>pvbus_s</code> 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.
<code>dbgen</code>	Signal	Slave	Invasive debug enable.
<code>errirq</code>	Signal	Master	Some async error was detected.
<code>evntcntoverflow[8]</code>	Signal	Master	The output interrupts of the event counters.
<code>niden</code>	Signal	Slave	Non-invasive debug enable.

Name	Protocol	Type	Description
pvbus_m[7]	PVBus	Master	The downstream 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.
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

For upstream port 0 determine if it is enabled or not w.r.t. snoop requests.

acchannelens1

Type

int

Default value

0x0

For upstream port 1 determine if it is enabled or not w.r.t. snoop requests.

acchannelens2

Type

int

Default value

0x0

For upstream port 2 determine if it is enabled or not w.r.t. snoop requests.

acchannelens3**Type**

int

Default value

0x0

For upstream port 3 determine if it is enabled or not w.r.t. snoop requests.

acchannelens4**Type**

int

Default value

0x0

For upstream port 4 determine if it is enabled or not w.r.t. snoop requests.

acchannelens5**Type**

int

Default value

0x0

For upstream port 5 determine if it is enabled or not w.r.t. snoop requests.

acchannelens6**Type**

int

Default value

0x0

For upstream port 6 determine if it is enabled or not w.r.t. snoop requests.

addr_width**Type**

int

Default value

0x28

The bit-width of the address that the CCI can accept.

cache_state_modelled**Type**

bool

Default value

0x1

Model the cache state.

dbgen**Type**

bool

Default value

0x1

Invasive debug enable. If true, enables the counting of PMU events.

enable_logger**Type**

bool

Default value

0x0

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 `ACCHANNLENSx` 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

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

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

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

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

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

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

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

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

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

""

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 3-913: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for CCN502

This model has the following Iris instances:

Table 3-914: 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

InstanceName	ComponentName
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
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 3-915: 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
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

InstanceName	ComponentName
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

CCN502 contains the following CADI targets:

- CCN5XX
- CCNCache

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 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.

Ports for CCN502

Table 3-916: Ports

Name	Protocol	Type	Description
<code>pvbus_m_hni[1]</code>	PVBus	Master	HNI downstream port.
<code>pvbus_m_snf[4]</code>	PVBus	Master	SNF downstream ports.
<code>pvbus_s_rnf[4]</code>	PVBus	Slave	RNF upstream ports.
<code>pvbus_s_rni[9]</code>	PVBus	Slave	RNI upstream ports
<code>reset_in</code>	Signal	Slave	Reset signal.

Parameters for CCN502

`acchannelen_rnf`

Type

int

Default value

0xf

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

`acchannelen_rni`

Type

int

Default value

0x1ff

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

Number of kilo bytes in cache.

cache_state_modelled**Type**

bool

Default value

0x1

Model the cache state.

ccn_cache.acchannelen_rnf**Type**

int

Default value

0xf

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

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

Number of kilo bytes in cache.

ccn_cache.cache_state_modelled**Type**

bool

Default value

0x1

Model the cache state.

ccn_cache.force_on_from_start**Type**

bool

Default value

0x0

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

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

Value for PERIPHBASE. Only bits [43:24] are used.

ccn_cache.reentrancy_support**Type**

string

Default value

env

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

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

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

enable_logger**Type**

bool

Default value

0x0

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

Number of SNF nodes present.

periphbase

Type
int

Default value
0x2c000000

Value for PERIPHBASE. Only bits [43:24] are used.

sbsx_bridge_present

Type
bool

Default value
0x1

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

systemaddrmap

Type
int

Default value
0x0

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"

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 3-917: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for CCN504

This model has the following Iris instances:

Table 3-918: 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
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

InstanceName	ComponentName
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

This model has the following MTI trace components:

Table 3-919: 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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
CCN504.ccn_router	PVBusMapper

CCN504 contains the following CADI targets:

- CCN5XX
- CCNCache

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.

Ports for CCN504

Table 3-920: 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

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

acchannelen_rni

Type

int

Default value

0x3ffff

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

Number of mega bytes in cache.

cache_state_modelled**Type**

bool

Default value

0x1

Model the cache state.

ccn_cache.acchannelen_rnf**Type**

int

Default value

0xf

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

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

Number of kilo bytes in cache.

ccn_cache.cache_state_modelled**Type**

bool

Default value

0x1

Model the cache state.

ccn_cache.force_on_from_start**Type**

bool

Default value

0x0

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

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

Value for PERIPHBASE. Only bits [43:24] are used.

ccn_cache.reentrancy_support**Type**

string

Default value

env

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

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

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

Disable sending error response when HNI receives cacheable access.

enable_logger**Type**

bool

Default value

0x0

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

Number of SNF nodes present.

periphbase**Type**

int

Default value

0x2c000000

Value for PERIPHBASE. Only bits [43:24] are used.

sbas_bridge_present**Type**

bool

Default value

0x1

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

sbsx_bridge_present**Type**

bool

Default value

0x1

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

systemaddrmap**Type**

int

Default value

0x0

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 3-921: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for CCN508

This model has the following Iris instances:

Table 3-922: 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
CCN508.ccn_cache.upstream[18]	PVBusSlave

InstanceName	ComponentName
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
CCN508.ccn_registers	CCNRegisterSet

InstanceName	ComponentName
CCN508.ccn_registers.bus_slave	PVBusSlave
CCN508.ccn_router	PVBusMapper

This model has the following MTI trace components:

Table 3-923: 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
CCN508.ccn_cache.upstream[34]	PVBusSlave

InstanceName	ComponentName
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

CCN508 contains the following CADI targets:

- CCN5XX
- CCNCache

About CCN508

CCN508 has six downstream ports:

- Four SNF ports for the memory controller.
- Two Acelite ports (HNI).



The parameters for the CCNCache subcomponent are not accessible in System Canvas.

Ports for CCN508

Table 3-924: Ports

Name	Protocol	Type	Description
pvbus_m_hni[2]	PVBus	Master	HNI downstream ports.
pvbus_m_snf[4]	PVBus	Master	SNF downstream 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.

Parameters for CCN508

acchannelen_rnf**Type**

int

Default value

0xf

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

acchannelen_rni**Type**

int

Default value

0x3ffff

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

Number of mega bytes in cache.

cache_state_modelled**Type**

bool

Default value

0x1

Model the cache state.

ccn_cache.acchannelen_rnf**Type**

int

Default value

0xf

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

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

Number of kilo bytes in cache.

ccn_cache.cache_state_modelled**Type**

bool

Default value

0x1

Model the cache state.

ccn_cache.force_on_from_start**Type**

bool

Default value

0x0

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

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

Value for PERIPHBASE. Only bits [43:24] are used.

ccn_cache.reentrancy_support

Type

string

Default value

env

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

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

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

Disable sending error response when HNI receives cacheable access.

enable_logger**Type**

bool

Default value

0x0

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

Number of SNF nodes present.

periphbase

Type
int

Default value
0x2c000000

Value for PERIPHBASE. Only bits [43:24] are used.

sbas_bridge_present

Type
bool

Default value
0x1

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

sbsx_bridge_present

Type
bool

Default value
0x1

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

systemaddrmap

Type
int

Default value
0x0

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 3-925: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for CCN512

This model has the following Iris instances:

Table 3-926: 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
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

InstanceName	ComponentName
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

This model has the following MTI trace components:

Table 3-927: 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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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

CCN512 contains the following CADI targets:

- CCN5XX
- CCNCache

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.

Ports for CCN512

Table 3-928: Ports

Name	Protocol	Type	Description
pvbus_m_hni[2]	PVBus	Master	HNI downstream ports.
pvbus_m_snf[4]	PVBus	Master	SNF downstream ports.
pvbus_s_rnf[12]	PVBus	Slave	RNF upstream ports.
pvbus_s_rni[24]	PVBus	Slave	RNI upstream ports.

Name	Protocol	Type	Description
reset_in	Signal	Slave	Reset signal.

Parameters for CCN512

acchannelen_rnf

Type

int

Default value

0xfff

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

acchannelen_rni

Type

int

Default value

0xffffffff

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

Number of mega bytes in cache.

cache_state_modelled

Type

bool

Default value

0x1

Model the cache state.

ccn_cache.acchannelen_rnf

Type

int

Default value

0xf

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

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

Number of kilo bytes in cache.

ccn_cache.cache_state_modelled**Type**

bool

Default value

0x1

Model the cache state.

ccn_cache.force_on_from_start**Type**

bool

Default value

0x0

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

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

Value for PERIPHBASE. Only bits [43:24] are used.

ccn_cache.reentrancy_support**Type**

string

Default value

env

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

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

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

enable_logger**Type**

bool

Default value

0x0

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

Number of SNF nodes present.

periphbase**Type**

int

Default value

0x2c000000

Value for PERIPHBASE. Only bits [43:24] are used.

sbsx_bridge_present**Type**

bool

Default value

0x1

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

systemaddrmap**Type**

int

Default value

0x0

Bitmap for 20 regions in CCN Interconnect. Every two bits describes the region type for corresponding region.

3.10.12 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 3-929: 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 23.

Iris and MTI instances for CI700

This model has the following Iris instances:

Table 3-930: 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	ExportTest.CI700.ci700_tag_cachemetadata_controller0
CI700.ci700_tag_cache.metadata_controller0.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller1	ExportTest.CI700.ci700_tag_cachemetadata_controller1
CI700.ci700_tag_cache.metadata_controller1.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller10	ExportTest.CI700.ci700_tag_cachemetadata_controller10
CI700.ci700_tag_cache.metadata_controller10.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller100	ExportTest.CI700.ci700_tag_cachemetadata_controller100

InstanceName	ComponentName
CI700.ci700_tag_cache. metadata_controller100.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller101	ExportTest.CI700.ci700_tag_cachemetadata_controller101
CI700.ci700_tag_cache. metadata_controller101.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller102	ExportTest.CI700.ci700_tag_cachemetadata_controller102
CI700.ci700_tag_cache. metadata_controller102.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller103	ExportTest.CI700.ci700_tag_cachemetadata_controller103
CI700.ci700_tag_cache. metadata_controller103.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller104	ExportTest.CI700.ci700_tag_cachemetadata_controller104
CI700.ci700_tag_cache. metadata_controller104.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller105	ExportTest.CI700.ci700_tag_cachemetadata_controller105
CI700.ci700_tag_cache. metadata_controller105.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller106	ExportTest.CI700.ci700_tag_cachemetadata_controller106
CI700.ci700_tag_cache. metadata_controller106.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller107	ExportTest.CI700.ci700_tag_cachemetadata_controller107
CI700.ci700_tag_cache. metadata_controller107.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller108	ExportTest.CI700.ci700_tag_cachemetadata_controller108
CI700.ci700_tag_cache. metadata_controller108.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller109	ExportTest.CI700.ci700_tag_cachemetadata_controller109
CI700.ci700_tag_cache. metadata_controller109.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller11	ExportTest.CI700.ci700_tag_cachemetadata_controller11
CI700.ci700_tag_cache. metadata_controller11.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller110	ExportTest.CI700.ci700_tag_cachemetadata_controller110
CI700.ci700_tag_cache. metadata_controller110.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller111	ExportTest.CI700.ci700_tag_cachemetadata_controller111
CI700.ci700_tag_cache. metadata_controller111.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller112	ExportTest.CI700.ci700_tag_cachemetadata_controller112
CI700.ci700_tag_cache. metadata_controller112.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller113	ExportTest.CI700.ci700_tag_cachemetadata_controller113
CI700.ci700_tag_cache. metadata_controller113.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller114	ExportTest.CI700.ci700_tag_cachemetadata_controller114
CI700.ci700_tag_cache. metadata_controller114.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller115	ExportTest.CI700.ci700_tag_cachemetadata_controller115
CI700.ci700_tag_cache. metadata_controller115.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller116	ExportTest.CI700.ci700_tag_cachemetadata_controller116
CI700.ci700_tag_cache. metadata_controller116.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller117	ExportTest.CI700.ci700_tag_cachemetadata_controller117
CI700.ci700_tag_cache. metadata_controller117.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller118	ExportTest.CI700.ci700_tag_cachemetadata_controller118
CI700.ci700_tag_cache. metadata_controller118.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller119	ExportTest.CI700.ci700_tag_cachemetadata_controller119
CI700.ci700_tag_cache. metadata_controller119.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller12	ExportTest.CI700.ci700_tag_cachemetadata_controller12
CI700.ci700_tag_cache. metadata_controller12.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller120	ExportTest.CI700.ci700_tag_cachemetadata_controller120
CI700.ci700_tag_cache. metadata_controller120.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller121	ExportTest.CI700.ci700_tag_cachemetadata_controller121
CI700.ci700_tag_cache. metadata_controller121.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller122	ExportTest.CI700.ci700_tag_cachemetadata_controller122
CI700.ci700_tag_cache. metadata_controller122.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller123	ExportTest.CI700.ci700_tag_cachemetadata_controller123
CI700.ci700_tag_cache. metadata_controller123.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller124	ExportTest.CI700.ci700_tag_cachemetadata_controller124
CI700.ci700_tag_cache. metadata_controller124.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller125	ExportTest.CI700.ci700_tag_cachemetadata_controller125
CI700.ci700_tag_cache. metadata_controller125.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller126	ExportTest.CI700.ci700_tag_cachemetadata_controller126
CI700.ci700_tag_cache. metadata_controller126.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller127	ExportTest.CI700.ci700_tag_cachemetadata_controller127

InstanceName	ComponentName
CI700.ci700_tag_cache. metadata_controller127.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller13	ExportTest.CI700.ci700_tag_cachemetadata_controller13
CI700.ci700_tag_cache. metadata_controller13.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller14	ExportTest.CI700.ci700_tag_cachemetadata_controller14
CI700.ci700_tag_cache. metadata_controller14.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller15	ExportTest.CI700.ci700_tag_cachemetadata_controller15
CI700.ci700_tag_cache. metadata_controller15.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller16	ExportTest.CI700.ci700_tag_cachemetadata_controller16
CI700.ci700_tag_cache. metadata_controller16.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller17	ExportTest.CI700.ci700_tag_cachemetadata_controller17
CI700.ci700_tag_cache. metadata_controller17.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller18	ExportTest.CI700.ci700_tag_cachemetadata_controller18
CI700.ci700_tag_cache. metadata_controller18.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller19	ExportTest.CI700.ci700_tag_cachemetadata_controller19
CI700.ci700_tag_cache. metadata_controller19.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller2	ExportTest.CI700.ci700_tag_cachemetadata_controller2
CI700.ci700_tag_cache. metadata_controller2.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller20	ExportTest.CI700.ci700_tag_cachemetadata_controller20
CI700.ci700_tag_cache. metadata_controller20.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller21	ExportTest.CI700.ci700_tag_cachemetadata_controller21
CI700.ci700_tag_cache. metadata_controller21.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller22	ExportTest.CI700.ci700_tag_cachemetadata_controller22
CI700.ci700_tag_cache. metadata_controller22.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller23	ExportTest.CI700.ci700_tag_cachemetadata_controller23
CI700.ci700_tag_cache. metadata_controller23.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller24	ExportTest.CI700.ci700_tag_cachemetadata_controller24
CI700.ci700_tag_cache. metadata_controller24.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller25	ExportTest.CI700.ci700_tag_cachemetadata_controller25
CI700.ci700_tag_cache. metadata_controller25.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller26	ExportTest.CI700.ci700_tag_cachemetadata_controller26
CI700.ci700_tag_cache. metadata_controller26.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller27	ExportTest.CI700.ci700_tag_cachemetadata_controller27
CI700.ci700_tag_cache. metadata_controller27.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller28	ExportTest.CI700.ci700_tag_cachemetadata_controller28
CI700.ci700_tag_cache. metadata_controller28.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller29	ExportTest.CI700.ci700_tag_cachemetadata_controller29
CI700.ci700_tag_cache. metadata_controller29.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller3	ExportTest.CI700.ci700_tag_cachemetadata_controller3
CI700.ci700_tag_cache. metadata_controller3.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller30	ExportTest.CI700.ci700_tag_cachemetadata_controller30
CI700.ci700_tag_cache. metadata_controller30.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller31	ExportTest.CI700.ci700_tag_cachemetadata_controller31
CI700.ci700_tag_cache. metadata_controller31.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller32	ExportTest.CI700.ci700_tag_cachemetadata_controller32
CI700.ci700_tag_cache. metadata_controller32.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller33	ExportTest.CI700.ci700_tag_cachemetadata_controller33
CI700.ci700_tag_cache. metadata_controller33.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller34	ExportTest.CI700.ci700_tag_cachemetadata_controller34
CI700.ci700_tag_cache. metadata_controller34.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller35	ExportTest.CI700.ci700_tag_cachemetadata_controller35
CI700.ci700_tag_cache. metadata_controller35.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller36	ExportTest.CI700.ci700_tag_cachemetadata_controller36
CI700.ci700_tag_cache. metadata_controller36.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller37	ExportTest.CI700.ci700_tag_cachemetadata_controller37
CI700.ci700_tag_cache. metadata_controller37.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller38	ExportTest.CI700.ci700_tag_cachemetadata_controller38
CI700.ci700_tag_cache. metadata_controller38.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller39	ExportTest.CI700.ci700_tag_cachemetadata_controller39

InstanceName	ComponentName
CI700.ci700_tag_cache. metadata_controller39.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller4	ExportTest.CI700.ci700_tag_cachemetadata_controller4
CI700.ci700_tag_cache. metadata_controller4.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller40	ExportTest.CI700.ci700_tag_cachemetadata_controller40
CI700.ci700_tag_cache. metadata_controller40.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller41	ExportTest.CI700.ci700_tag_cachemetadata_controller41
CI700.ci700_tag_cache. metadata_controller41.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller42	ExportTest.CI700.ci700_tag_cachemetadata_controller42
CI700.ci700_tag_cache. metadata_controller42.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller43	ExportTest.CI700.ci700_tag_cachemetadata_controller43
CI700.ci700_tag_cache. metadata_controller43.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller44	ExportTest.CI700.ci700_tag_cachemetadata_controller44
CI700.ci700_tag_cache. metadata_controller44.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller45	ExportTest.CI700.ci700_tag_cachemetadata_controller45
CI700.ci700_tag_cache. metadata_controller45.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller46	ExportTest.CI700.ci700_tag_cachemetadata_controller46
CI700.ci700_tag_cache. metadata_controller46.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller47	ExportTest.CI700.ci700_tag_cachemetadata_controller47
CI700.ci700_tag_cache. metadata_controller47.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller48	ExportTest.CI700.ci700_tag_cachemetadata_controller48
CI700.ci700_tag_cache. metadata_controller48.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller49	ExportTest.CI700.ci700_tag_cachemetadata_controller49
CI700.ci700_tag_cache. metadata_controller49.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller5	ExportTest.CI700.ci700_tag_cachemetadata_controller5
CI700.ci700_tag_cache. metadata_controller5.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller50	ExportTest.CI700.ci700_tag_cachemetadata_controller50
CI700.ci700_tag_cache. metadata_controller50.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller51	ExportTest.CI700.ci700_tag_cachemetadata_controller51
CI700.ci700_tag_cache. metadata_controller51.MetadataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller52	ExportTest.CI700.ci700_tag_cachemetadata_controller52
CI700.ci700_tag_cache. metadata_controller52.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller53	ExportTest.CI700.ci700_tag_cachemetadata_controller53
CI700.ci700_tag_cache. metadata_controller53.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller54	ExportTest.CI700.ci700_tag_cachemetadata_controller54
CI700.ci700_tag_cache. metadata_controller54.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller55	ExportTest.CI700.ci700_tag_cachemetadata_controller55
CI700.ci700_tag_cache. metadata_controller55.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller56	ExportTest.CI700.ci700_tag_cachemetadata_controller56
CI700.ci700_tag_cache. metadata_controller56.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller57	ExportTest.CI700.ci700_tag_cachemetadata_controller57
CI700.ci700_tag_cache. metadata_controller57.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller58	ExportTest.CI700.ci700_tag_cachemetadata_controller58
CI700.ci700_tag_cache. metadata_controller58.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller59	ExportTest.CI700.ci700_tag_cachemetadata_controller59
CI700.ci700_tag_cache. metadata_controller59.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller6	ExportTest.CI700.ci700_tag_cachemetadata_controller6
CI700.ci700_tag_cache. metadata_controller6.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller60	ExportTest.CI700.ci700_tag_cachemetadata_controller60
CI700.ci700_tag_cache. metadata_controller60.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller61	ExportTest.CI700.ci700_tag_cachemetadata_controller61
CI700.ci700_tag_cache. metadata_controller61.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller62	ExportTest.CI700.ci700_tag_cachemetadata_controller62
CI700.ci700_tag_cache. metadata_controller62.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller63	ExportTest.CI700.ci700_tag_cachemetadata_controller63
CI700.ci700_tag_cache. metadata_controller63.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller64	ExportTest.CI700.ci700_tag_cachemetadata_controller64
CI700.ci700_tag_cache. metadata_controller64.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller65	ExportTest.CI700.ci700_tag_cachemetadata_controller65

InstanceName	ComponentName
CI700.ci700_tag_cache. metadata_controller65.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller66	ExportTest.CI700.ci700_tag_cachemetadata_controller66
CI700.ci700_tag_cache. metadata_controller66.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller67	ExportTest.CI700.ci700_tag_cachemetadata_controller67
CI700.ci700_tag_cache. metadata_controller67.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller68	ExportTest.CI700.ci700_tag_cachemetadata_controller68
CI700.ci700_tag_cache. metadata_controller68.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller69	ExportTest.CI700.ci700_tag_cachemetadata_controller69
CI700.ci700_tag_cache. metadata_controller69.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller7	ExportTest.CI700.ci700_tag_cachemetadata_controller7
CI700.ci700_tag_cache. metadata_controller7.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller70	ExportTest.CI700.ci700_tag_cachemetadata_controller70
CI700.ci700_tag_cache. metadata_controller70.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller71	ExportTest.CI700.ci700_tag_cachemetadata_controller71
CI700.ci700_tag_cache. metadata_controller71.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller72	ExportTest.CI700.ci700_tag_cachemetadata_controller72
CI700.ci700_tag_cache. metadata_controller72.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller73	ExportTest.CI700.ci700_tag_cachemetadata_controller73
CI700.ci700_tag_cache. metadata_controller73.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller74	ExportTest.CI700.ci700_tag_cachemetadata_controller74
CI700.ci700_tag_cache. metadata_controller74.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller75	ExportTest.CI700.ci700_tag_cachemetadata_controller75
CI700.ci700_tag_cache. metadata_controller75.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller76	ExportTest.CI700.ci700_tag_cachemetadata_controller76
CI700.ci700_tag_cache. metadata_controller76.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller77	ExportTest.CI700.ci700_tag_cachemetadata_controller77
CI700.ci700_tag_cache. metadata_controller77.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller78	ExportTest.CI700.ci700_tag_cachemetadata_controller78
CI700.ci700_tag_cache. metadata_controller78.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller79	ExportTest.CI700.ci700_tag_cachemetadata_controller79
CI700.ci700_tag_cache. metadata_controller79.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller8	ExportTest.CI700.ci700_tag_cachemetadata_controller8
CI700.ci700_tag_cache. metadata_controller8.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller80	ExportTest.CI700.ci700_tag_cachemetadata_controller80
CI700.ci700_tag_cache. metadata_controller80.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller81	ExportTest.CI700.ci700_tag_cachemetadata_controller81
CI700.ci700_tag_cache. metadata_controller81.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller82	ExportTest.CI700.ci700_tag_cachemetadata_controller82
CI700.ci700_tag_cache. metadata_controller82.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller83	ExportTest.CI700.ci700_tag_cachemetadata_controller83
CI700.ci700_tag_cache. metadata_controller83.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller84	ExportTest.CI700.ci700_tag_cachemetadata_controller84
CI700.ci700_tag_cache. metadata_controller84.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller85	ExportTest.CI700.ci700_tag_cachemetadata_controller85
CI700.ci700_tag_cache. metadata_controller85.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller86	ExportTest.CI700.ci700_tag_cachemetadata_controller86
CI700.ci700_tag_cache. metadata_controller86.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller87	ExportTest.CI700.ci700_tag_cachemetadata_controller87
CI700.ci700_tag_cache. metadata_controller87.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller88	ExportTest.CI700.ci700_tag_cachemetadata_controller88
CI700.ci700_tag_cache. metadata_controller88.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller89	ExportTest.CI700.ci700_tag_cachemetadata_controller89
CI700.ci700_tag_cache. metadata_controller89.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller9	ExportTest.CI700.ci700_tag_cachemetadata_controller9
CI700.ci700_tag_cache. metadata_controller9.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller90	ExportTest.CI700.ci700_tag_cachemetadata_controller90
CI700.ci700_tag_cache. metadata_controller90.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller91	ExportTest.CI700.ci700_tag_cachemetadata_controller91

InstanceName	ComponentName
CI700.ci700_tag_cache. metadata_controller91.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller92	ExportTest.CI700.ci700_tag_cachemetadata_controller92
CI700.ci700_tag_cache. metadata_controller92.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller93	ExportTest.CI700.ci700_tag_cachemetadata_controller93
CI700.ci700_tag_cache. metadata_controller93.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller94	ExportTest.CI700.ci700_tag_cachemetadata_controller94
CI700.ci700_tag_cache. metadata_controller94.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller95	ExportTest.CI700.ci700_tag_cachemetadata_controller95
CI700.ci700_tag_cache. metadata_controller95.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller96	ExportTest.CI700.ci700_tag_cachemetadata_controller96
CI700.ci700_tag_cache. metadata_controller96.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller97	ExportTest.CI700.ci700_tag_cachemetadata_controller97
CI700.ci700_tag_cache. metadata_controller97.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller98	ExportTest.CI700.ci700_tag_cachemetadata_controller98
CI700.ci700_tag_cache. metadata_controller98.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller99	ExportTest.CI700.ci700_tag_cachemetadata_controller99
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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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 3-931: 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	ExportTest.CI700.ci700_tag_cachemetadata_controller0
CI700.ci700_tag_cache.metadata_controller0.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller1	ExportTest.CI700.ci700_tag_cachemetadata_controller1
CI700.ci700_tag_cache.metadata_controller1.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller10	ExportTest.CI700.ci700_tag_cachemetadata_controller10
CI700.ci700_tag_cache.metadata_controller10.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller100	ExportTest.CI700.ci700_tag_cachemetadata_controller100
CI700.ci700_tag_cache.metadata_controller100.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller101	ExportTest.CI700.ci700_tag_cachemetadata_controller101
CI700.ci700_tag_cache.metadata_controller101.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller102	ExportTest.CI700.ci700_tag_cachemetadata_controller102
CI700.ci700_tag_cache.metadata_controller102.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller103	ExportTest.CI700.ci700_tag_cachemetadata_controller103
CI700.ci700_tag_cache.metadata_controller103.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller104	ExportTest.CI700.ci700_tag_cachemetadata_controller104
CI700.ci700_tag_cache.metadata_controller104.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller105	ExportTest.CI700.ci700_tag_cachemetadata_controller105

InstanceName	ComponentName
CI700.ci700_tag_cache. metadata_controller105.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller106	ExportTest.CI700.ci700_tag_cachemetadata_controller106
CI700.ci700_tag_cache. metadata_controller106.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller107	ExportTest.CI700.ci700_tag_cachemetadata_controller107
CI700.ci700_tag_cache. metadata_controller107.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller108	ExportTest.CI700.ci700_tag_cachemetadata_controller108
CI700.ci700_tag_cache. metadata_controller108.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller109	ExportTest.CI700.ci700_tag_cachemetadata_controller109
CI700.ci700_tag_cache. metadata_controller109.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller11	ExportTest.CI700.ci700_tag_cachemetadata_controller11
CI700.ci700_tag_cache. metadata_controller11.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller110	ExportTest.CI700.ci700_tag_cachemetadata_controller110
CI700.ci700_tag_cache. metadata_controller110.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller111	ExportTest.CI700.ci700_tag_cachemetadata_controller111
CI700.ci700_tag_cache. metadata_controller111.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller112	ExportTest.CI700.ci700_tag_cachemetadata_controller112
CI700.ci700_tag_cache. metadata_controller112.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller113	ExportTest.CI700.ci700_tag_cachemetadata_controller113
CI700.ci700_tag_cache. metadata_controller113.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller114	ExportTest.CI700.ci700_tag_cachemetadata_controller114
CI700.ci700_tag_cache. metadata_controller114.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller115	ExportTest.CI700.ci700_tag_cachemetadata_controller115
CI700.ci700_tag_cache. metadata_controller115.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller116	ExportTest.CI700.ci700_tag_cachemetadata_controller116
CI700.ci700_tag_cache. metadata_controller116.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller117	ExportTest.CI700.ci700_tag_cachemetadata_controller117
CI700.ci700_tag_cache. metadata_controller117.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller118	ExportTest.CI700.ci700_tag_cachemetadata_controller118
CI700.ci700_tag_cache. metadata_controller118.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller119	ExportTest.CI700.ci700_tag_cachemetadata_controller119
CI700.ci700_tag_cache. metadata_controller119.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller12	ExportTest.CI700.ci700_tag_cachemetadata_controller12
CI700.ci700_tag_cache. metadata_controller12.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller120	ExportTest.CI700.ci700_tag_cachemetadata_controller120
CI700.ci700_tag_cache. metadata_controller120.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller121	ExportTest.CI700.ci700_tag_cachemetadata_controller121
CI700.ci700_tag_cache. metadata_controller121.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller122	ExportTest.CI700.ci700_tag_cachemetadata_controller122
CI700.ci700_tag_cache. metadata_controller122.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller123	ExportTest.CI700.ci700_tag_cachemetadata_controller123
CI700.ci700_tag_cache. metadata_controller123.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller124	ExportTest.CI700.ci700_tag_cachemetadata_controller124
CI700.ci700_tag_cache. metadata_controller124.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller125	ExportTest.CI700.ci700_tag_cachemetadata_controller125
CI700.ci700_tag_cache. metadata_controller125.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller126	ExportTest.CI700.ci700_tag_cachemetadata_controller126
CI700.ci700_tag_cache. metadata_controller126.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller127	ExportTest.CI700.ci700_tag_cachemetadata_controller127
CI700.ci700_tag_cache. metadata_controller127.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller13	ExportTest.CI700.ci700_tag_cachemetadata_controller13
CI700.ci700_tag_cache. metadata_controller13.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller14	ExportTest.CI700.ci700_tag_cachemetadata_controller14
CI700.ci700_tag_cache. metadata_controller14.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller15	ExportTest.CI700.ci700_tag_cachemetadata_controller15
CI700.ci700_tag_cache. metadata_controller15.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller16	ExportTest.CI700.ci700_tag_cachemetadata_controller16
CI700.ci700_tag_cache. metadata_controller16.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller17	ExportTest.CI700.ci700_tag_cachemetadata_controller17

InstanceName	ComponentName
CI700.ci700_tag_cache. metadata_controller17.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller18	ExportTest.CI700.ci700_tag_cachemetadata_controller18
CI700.ci700_tag_cache. metadata_controller18.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller19	ExportTest.CI700.ci700_tag_cachemetadata_controller19
CI700.ci700_tag_cache. metadata_controller19.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller2	ExportTest.CI700.ci700_tag_cachemetadata_controller2
CI700.ci700_tag_cache. metadata_controller2.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller20	ExportTest.CI700.ci700_tag_cachemetadata_controller20
CI700.ci700_tag_cache. metadata_controller20.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller21	ExportTest.CI700.ci700_tag_cachemetadata_controller21
CI700.ci700_tag_cache. metadata_controller21.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller22	ExportTest.CI700.ci700_tag_cachemetadata_controller22
CI700.ci700_tag_cache. metadata_controller22.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller23	ExportTest.CI700.ci700_tag_cachemetadata_controller23
CI700.ci700_tag_cache. metadata_controller23.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller24	ExportTest.CI700.ci700_tag_cachemetadata_controller24
CI700.ci700_tag_cache. metadata_controller24.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller25	ExportTest.CI700.ci700_tag_cachemetadata_controller25
CI700.ci700_tag_cache. metadata_controller25.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller26	ExportTest.CI700.ci700_tag_cachemetadata_controller26
CI700.ci700_tag_cache. metadata_controller26.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller27	ExportTest.CI700.ci700_tag_cachemetadata_controller27
CI700.ci700_tag_cache. metadata_controller27.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller28	ExportTest.CI700.ci700_tag_cachemetadata_controller28
CI700.ci700_tag_cache. metadata_controller28.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller29	ExportTest.CI700.ci700_tag_cachemetadata_controller29
CI700.ci700_tag_cache. metadata_controller29.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller3	ExportTest.CI700.ci700_tag_cachemetadata_controller3
CI700.ci700_tag_cache. metadata_controller3.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller30	ExportTest.CI700.ci700_tag_cachemetadata_controller30
CI700.ci700_tag_cache. metadata_controller30.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller31	ExportTest.CI700.ci700_tag_cachemetadata_controller31
CI700.ci700_tag_cache. metadata_controller31.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller32	ExportTest.CI700.ci700_tag_cachemetadata_controller32
CI700.ci700_tag_cache. metadata_controller32.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller33	ExportTest.CI700.ci700_tag_cachemetadata_controller33
CI700.ci700_tag_cache. metadata_controller33.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller34	ExportTest.CI700.ci700_tag_cachemetadata_controller34
CI700.ci700_tag_cache. metadata_controller34.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller35	ExportTest.CI700.ci700_tag_cachemetadata_controller35
CI700.ci700_tag_cache. metadata_controller35.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller36	ExportTest.CI700.ci700_tag_cachemetadata_controller36
CI700.ci700_tag_cache. metadata_controller36.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller37	ExportTest.CI700.ci700_tag_cachemetadata_controller37
CI700.ci700_tag_cache. metadata_controller37.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller38	ExportTest.CI700.ci700_tag_cachemetadata_controller38
CI700.ci700_tag_cache. metadata_controller38.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller39	ExportTest.CI700.ci700_tag_cachemetadata_controller39
CI700.ci700_tag_cache. metadata_controller39.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller4	ExportTest.CI700.ci700_tag_cachemetadata_controller4
CI700.ci700_tag_cache. metadata_controller4.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller40	ExportTest.CI700.ci700_tag_cachemetadata_controller40
CI700.ci700_tag_cache. metadata_controller40.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller41	ExportTest.CI700.ci700_tag_cachemetadata_controller41
CI700.ci700_tag_cache. metadata_controller41.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller42	ExportTest.CI700.ci700_tag_cachemetadata_controller42
CI700.ci700_tag_cache. metadata_controller42.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller43	ExportTest.CI700.ci700_tag_cachemetadata_controller43

InstanceName	ComponentName
CI700.ci700_tag_cache. metadata_controller43.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller44	ExportTest.CI700.ci700_tag_cachemetadata_controller44
CI700.ci700_tag_cache. metadata_controller44.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller45	ExportTest.CI700.ci700_tag_cachemetadata_controller45
CI700.ci700_tag_cache. metadata_controller45.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller46	ExportTest.CI700.ci700_tag_cachemetadata_controller46
CI700.ci700_tag_cache. metadata_controller46.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller47	ExportTest.CI700.ci700_tag_cachemetadata_controller47
CI700.ci700_tag_cache. metadata_controller47.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller48	ExportTest.CI700.ci700_tag_cachemetadata_controller48
CI700.ci700_tag_cache. metadata_controller48.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller49	ExportTest.CI700.ci700_tag_cachemetadata_controller49
CI700.ci700_tag_cache. metadata_controller49.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller5	ExportTest.CI700.ci700_tag_cachemetadata_controller5
CI700.ci700_tag_cache. metadata_controller5.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller50	ExportTest.CI700.ci700_tag_cachemetadata_controller50
CI700.ci700_tag_cache. metadata_controller50.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller51	ExportTest.CI700.ci700_tag_cachemetadata_controller51
CI700.ci700_tag_cache. metadata_controller51.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller52	ExportTest.CI700.ci700_tag_cachemetadata_controller52
CI700.ci700_tag_cache. metadata_controller52.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller53	ExportTest.CI700.ci700_tag_cachemetadata_controller53
CI700.ci700_tag_cache. metadata_controller53.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller54	ExportTest.CI700.ci700_tag_cachemetadata_controller54
CI700.ci700_tag_cache. metadata_controller54.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller55	ExportTest.CI700.ci700_tag_cachemetadata_controller55
CI700.ci700_tag_cache. metadata_controller55.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller56	ExportTest.CI700.ci700_tag_cachemetadata_controller56
CI700.ci700_tag_cache. metadata_controller56.MetadataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller57	ExportTest.CI700.ci700_tag_cachemetadata_controller57
CI700.ci700_tag_cache. metadata_controller57.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller58	ExportTest.CI700.ci700_tag_cachemetadata_controller58
CI700.ci700_tag_cache. metadata_controller58.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller59	ExportTest.CI700.ci700_tag_cachemetadata_controller59
CI700.ci700_tag_cache. metadata_controller59.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller6	ExportTest.CI700.ci700_tag_cachemetadata_controller6
CI700.ci700_tag_cache. metadata_controller6.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller60	ExportTest.CI700.ci700_tag_cachemetadata_controller60
CI700.ci700_tag_cache. metadata_controller60.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller61	ExportTest.CI700.ci700_tag_cachemetadata_controller61
CI700.ci700_tag_cache. metadata_controller61.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller62	ExportTest.CI700.ci700_tag_cachemetadata_controller62
CI700.ci700_tag_cache. metadata_controller62.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller63	ExportTest.CI700.ci700_tag_cachemetadata_controller63
CI700.ci700_tag_cache. metadata_controller63.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller64	ExportTest.CI700.ci700_tag_cachemetadata_controller64
CI700.ci700_tag_cache. metadata_controller64.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller65	ExportTest.CI700.ci700_tag_cachemetadata_controller65
CI700.ci700_tag_cache. metadata_controller65.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller66	ExportTest.CI700.ci700_tag_cachemetadata_controller66
CI700.ci700_tag_cache. metadata_controller66.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller67	ExportTest.CI700.ci700_tag_cachemetadata_controller67
CI700.ci700_tag_cache. metadata_controller67.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller68	ExportTest.CI700.ci700_tag_cachemetadata_controller68
CI700.ci700_tag_cache. metadata_controller68.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller69	ExportTest.CI700.ci700_tag_cachemetadata_controller69
CI700.ci700_tag_cache. metadata_controller69.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller7	ExportTest.CI700.ci700_tag_cachemetadata_controller7

InstanceName	ComponentName
CI700.ci700_tag_cache. metadata_controller7.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller70	ExportTest.CI700.ci700_tag_cachemetadata_controller70
CI700.ci700_tag_cache. metadata_controller70.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller71	ExportTest.CI700.ci700_tag_cachemetadata_controller71
CI700.ci700_tag_cache. metadata_controller71.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller72	ExportTest.CI700.ci700_tag_cachemetadata_controller72
CI700.ci700_tag_cache. metadata_controller72.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller73	ExportTest.CI700.ci700_tag_cachemetadata_controller73
CI700.ci700_tag_cache. metadata_controller73.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller74	ExportTest.CI700.ci700_tag_cachemetadata_controller74
CI700.ci700_tag_cache. metadata_controller74.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller75	ExportTest.CI700.ci700_tag_cachemetadata_controller75
CI700.ci700_tag_cache. metadata_controller75.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller76	ExportTest.CI700.ci700_tag_cachemetadata_controller76
CI700.ci700_tag_cache. metadata_controller76.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller77	ExportTest.CI700.ci700_tag_cachemetadata_controller77
CI700.ci700_tag_cache. metadata_controller77.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller78	ExportTest.CI700.ci700_tag_cachemetadata_controller78
CI700.ci700_tag_cache. metadata_controller78.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller79	ExportTest.CI700.ci700_tag_cachemetadata_controller79
CI700.ci700_tag_cache. metadata_controller79.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller8	ExportTest.CI700.ci700_tag_cachemetadata_controller8
CI700.ci700_tag_cache. metadata_controller8.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller80	ExportTest.CI700.ci700_tag_cachemetadata_controller80
CI700.ci700_tag_cache. metadata_controller80.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller81	ExportTest.CI700.ci700_tag_cachemetadata_controller81
CI700.ci700_tag_cache. metadata_controller81.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller82	ExportTest.CI700.ci700_tag_cachemetadata_controller82
CI700.ci700_tag_cache. metadata_controller82.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller83	ExportTest.CI700.ci700_tag_cachemetadata_controller83
CI700.ci700_tag_cache. metadata_controller83.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller84	ExportTest.CI700.ci700_tag_cachemetadata_controller84
CI700.ci700_tag_cache. metadata_controller84.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller85	ExportTest.CI700.ci700_tag_cachemetadata_controller85
CI700.ci700_tag_cache. metadata_controller85.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller86	ExportTest.CI700.ci700_tag_cachemetadata_controller86
CI700.ci700_tag_cache. metadata_controller86.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller87	ExportTest.CI700.ci700_tag_cachemetadata_controller87
CI700.ci700_tag_cache. metadata_controller87.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller88	ExportTest.CI700.ci700_tag_cachemetadata_controller88
CI700.ci700_tag_cache. metadata_controller88.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller89	ExportTest.CI700.ci700_tag_cachemetadata_controller89
CI700.ci700_tag_cache. metadata_controller89.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller9	ExportTest.CI700.ci700_tag_cachemetadata_controller9
CI700.ci700_tag_cache. metadata_controller9.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller90	ExportTest.CI700.ci700_tag_cachemetadata_controller90
CI700.ci700_tag_cache. metadata_controller90.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller91	ExportTest.CI700.ci700_tag_cachemetadata_controller91
CI700.ci700_tag_cache. metadata_controller91.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller92	ExportTest.CI700.ci700_tag_cachemetadata_controller92
CI700.ci700_tag_cache. metadata_controller92.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller93	ExportTest.CI700.ci700_tag_cachemetadata_controller93
CI700.ci700_tag_cache. metadata_controller93.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller94	ExportTest.CI700.ci700_tag_cachemetadata_controller94
CI700.ci700_tag_cache. metadata_controller94.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller95	ExportTest.CI700.ci700_tag_cachemetadata_controller95
CI700.ci700_tag_cache. metadata_controller95.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller96	ExportTest.CI700.ci700_tag_cachemetadata_controller96

InstanceName	ComponentName
CI700.ci700_tag_cache. metadata_controller96.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller97	ExportTest.CI700.ci700_tag_cachemetadata_controller97
CI700.ci700_tag_cache. metadata_controller97.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller98	ExportTest.CI700.ci700_tag_cachemetadata_controller98
CI700.ci700_tag_cache. metadata_controller98.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller99	ExportTest.CI700.ci700_tag_cachemetadata_controller99
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
CI700.ci700_tag_cache.remapper123	PVBusMapper

InstanceName	ComponentName
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
CI700.ci700_tag_cache.remapper44	PVBusMapper

InstanceName	ComponentName
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
CI700.ci700_tag_cache.remapper8	PVBusMapper

InstanceName	ComponentName
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

CI700 contains the following CADI targets:

- CI700
- CMN_TAG_CACHE

About CI700

- 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 `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*_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 `0b00000`. 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 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 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.
 - 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.

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.

Ports for CI700

Table 3-932: Ports

Name	Protocol	Type	Description
event_clusters	Signal	Peer	CPU event communication signal from the clusters.
pvbus_m_hni[4]	PVBus	Master	HNI downstream ports.
pvbus_m_snf[8]	PVBus	Master	SNF downstream port.
pvbus_s_apb	PVBus	Slave	APB interface port.
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 CI700

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 `ffff`

bypass_tag_cache

Type
bool

Default value
0x0

If true, CI700 will bypass the tag cache component which provides the MTE support.

cache_state_modelled

Type
bool

Default value
0x0

Model the cache state.

ci700_tag_cache.metadata_controller.init_value

Type
int

Default value
0xd

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

""

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

""

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

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

""

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 ≥ 64 bytes and a power of 2, defaulting to 4KiB. The maximum block size supported is 4KiB. The carveout region size must be ≥ 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

""

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

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. THIS PARAMETER HAS NO FUNCTIONALITY AT THE MOMENT.

ci700_tag_cache.metadata_controller.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.

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.



This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

`enable_logger`

Type

bool

Default value

0x0

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

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

""

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_cm_n_ccix_config**Type**

bool

Default value

0x0

Print information about the CCIX configuration.

print_cm_n_config**Type**

bool

Default value

0x0

Print the mesh topology and children pointers acquired from the YML file.

revision**Type**

string

Default value

"rOp0"

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

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

Does the top-level YML file describe node-addresses ?.

3.10.13 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 3-933: IP revisions support

Revision	Quality level
r1p1	Full support

Revision	Quality level
r3p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for CMN600

This model has the following Iris instances:

Table 3-934: 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 3-935: CMN600 MTI 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

CMN600 contains the following CADI targets:

- CMN600
- CMN600Cache

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

0KB where `HN*_SLC_NUM_WAYS_PARAM=16`



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 `0b00000`. 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 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 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.

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.

Ports for CMN600

Table 3-936: Ports

Name	Protocol	Type	Description
<code>event_clusters</code>	Signal	Peer	CPU event communication signal from the clusters.
<code>event_downstream_link_signal[6]</code>	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.
<code>event_upstream_link_signal[6]</code>	Signal	Slave	Event from the Hub towards the CMN
<code>pvbust_m_cml</code>	PVBus	Master	CML downstream ports
<code>pvbust_m_cml_cfg</code>	PVBus	Master	CML downstream hub configuration port
<code>pvbust_m_hni[8]</code>	PVBus	Master	HNI downstream ports.
<code>pvbust_m_snf[16]</code>	PVBus	Master	SNF downstream port.
<code>pvbust_s_cml</code>	PVBus	Slave	CML upstream ports

Name	Protocol	Type	Description
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 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

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

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

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

Force all RNSAMs to be internal independently of the mesh topology.

mesh_config_file**Type**

string

Default value

""

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

Print information about the CCIX configuration.

print_cmn_config**Type**

bool

Default value

0x0

Print the mesh topology and children pointers acquired from the YML 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

supress entire banner

1

suppress config file

2+

show full banner

skip_cmn_config_check**Type**

bool

Default value

0x0

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.

3.10.14 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 3-937: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for CMN600AE

This model has the following Iris instances:

Table 3-938: 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

InstanceName	ComponentName
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
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 3-939: 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

InstanceName	ComponentName
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
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

CMN600AE contains the following CADI targets:

- CMN600AE

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 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 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_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 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:

-8OKB when `HN*_SLC_NUM_WAYS_PARAM=16`

This value is not supported by the model.

-2128KB 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 `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.
- 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 `0b00000`. 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 `mxp_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.

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.

Ports for CMN600AE

Table 3-940: 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"



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

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

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

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.

fmw_key**Type**

uint8_t

Default value

0

Por_fm_key register value is checked against this key.

force_rnsam_internal**Type**

bool

Default value

0x1

Force all RNSAMs to be internal independently of the mesh topology.

mesh_config_file**Type**

string

Default value

""

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

Print information about the CCIX configuration.

print_cmn_config**Type**

bool

Default value

0x0

Print the mesh topology and children pointers acquired from the YML 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

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.

3.10.15 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 3-941: 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 23.

Iris and MTI instances for CMN600CMLHub

This model has the following Iris instances:

Table 3-942: 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 3-943: 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

CMN600CMLHub contains the following CADI targets:

- CMN600CMLHub

Ports for CMN600CMLHub

Table 3-944: 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.
pdbus_m[4]	PVBus	Master	Downstream CCIX port.
pdbus_s[4]	PVBus	Slave	Upstream CCIX ports.
pdbus_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

Model the cache state.

enable_logger**Type**

bool

Default value

0x0

Enable PVBusLoggers for the downstream ports in the CMN model.

3.10.16 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 3-945: IP revisions support

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for CMN650

This model has the following Iris instances:

Table 3-946: 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

InstanceName	ComponentName
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
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 3-947: CMN650 MTI instances

InstanceName	ComponentName
CMN650	CMN650
CMN650.bus_slave_ocm_NS	PVBusSlave

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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

CMN650 contains the following CADI targets:

- CMN650

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 `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 CMN650 with two `RN-D` nodes, one `RN-I` node, and given each `RN-I` or `RN-D` node controls three interface ports, `pvbuss_rni[0-2]` maps to `RND0`, `pvbuss_rni[3-5]` maps to `RND1` and `pvbuss_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`



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 0b00000. 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 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.
- GLC 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 `HN_F_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 `HN_F_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.

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.

Ports for CMN650

Table 3-948: 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
pvbuse_m_cml	PVBus	Master	CML downstream ports
pvbuse_m_cml_cfg	PVBus	Master	CML downstream hub configuration port
pvbuse_m_hni[16]	PVBus	Master	HNI downstream ports.
pvbuse_m_snf[40]	PVBus	Master	SNF downstream port.
pvbuse_s_apb	PVBus	Slave	APB interface port.
pvbuse_s_cml	PVBus	Slave	CML upstream ports
pvbuse_s_rnf[64]	PVBus	Slave	RNF upstream ports.
pvbuse_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"



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

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.



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

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

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

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

""

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

Print information about the CCIX configuration.

print_cmn_config**Type**

bool

Default value

0x0

Print the mesh topology and children pointers acquired from the YML 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

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.

3.10.17 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 3-949: IP revisions support

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for CMN650R2

This model has the following Iris instances:

Table 3-950: 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

InstanceName	ComponentName
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

This model has the following MTI trace components:

Table 3-951: 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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
CMN650R2.snf_mapper	PVBusMapper

CMN650R2 contains the following CADI targets:

- CMN650R2

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 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 `pvbush_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, `pvbush_s_rni[0-2]` maps to RND0, `pvbush_s_rni[3-5]` maps to RND1 and `pvbush_s_rni[6-8]` maps to RNI0.

Similarly, SN-F and SBSX nodes are mapped to `pvbush_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 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

11MB 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 `0b00000`. 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 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 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.
- 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.

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.

Ports for CMN650R2

Table 3-952: 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
pvbust_m_cml	PVBus	Master	CML downstream ports
pvbust_m_cml_cfg	PVBus	Master	CML downstream hub configuration port
pvbust_m_hni[16]	PVBus	Master	HNI downstream ports.
pvbust_m_snf[40]	PVBus	Master	SNF downstream port.
pvbust_s_apb	PVBus	Slave	APB interface port.
pvbust_s_cml	PVBus	Slave	CML upstream ports
pvbust_s_rnf[256]	PVBus	Slave	RNF upstream ports.
pvbust_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"



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

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.



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

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

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

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

""

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

Print information about the CCIX configuration.

print_cmn_config**Type**

bool

Default value

0x0

Print the mesh topology and children pointers acquired from the YML 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

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 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 3-953: 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 23.

Iris and MTI instances for CMN700

This model has the following Iris instances:

Table 3-954: 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
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

InstanceName	ComponentName
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	ExportTest.CMN700.cmn700_tag_cachemetadata_controller0
CMN700.cmn700_tag_cache. metadata_controller0.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller1	ExportTest.CMN700.cmn700_tag_cachemetadata_controller1
CMN700.cmn700_tag_cache. metadata_controller1.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller10	ExportTest.CMN700.cmn700_tag_cachemetadata_controller10
CMN700.cmn700_tag_cache. metadata_controller10.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller100	ExportTest.CMN700.cmn700_tag_cachemetadata_controller100
CMN700.cmn700_tag_cache. metadata_controller100.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller101	ExportTest.CMN700.cmn700_tag_cachemetadata_controller101
CMN700.cmn700_tag_cache. metadata_controller101.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller102	ExportTest.CMN700.cmn700_tag_cachemetadata_controller102
CMN700.cmn700_tag_cache. metadata_controller102.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller103	ExportTest.CMN700.cmn700_tag_cachemetadata_controller103

InstanceName	ComponentName
CMN700.cmn700_tag_cache. metadata_controller103.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller104	ExportTest.CMN700.cmn700_tag_cachemetadata_controller104
CMN700.cmn700_tag_cache. metadata_controller104.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller105	ExportTest.CMN700.cmn700_tag_cachemetadata_controller105
CMN700.cmn700_tag_cache. metadata_controller105.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller106	ExportTest.CMN700.cmn700_tag_cachemetadata_controller106
CMN700.cmn700_tag_cache. metadata_controller106.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller107	ExportTest.CMN700.cmn700_tag_cachemetadata_controller107
CMN700.cmn700_tag_cache. metadata_controller107.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller108	ExportTest.CMN700.cmn700_tag_cachemetadata_controller108
CMN700.cmn700_tag_cache. metadata_controller108.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller109	ExportTest.CMN700.cmn700_tag_cachemetadata_controller109
CMN700.cmn700_tag_cache. metadata_controller109.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller11	ExportTest.CMN700.cmn700_tag_cachemetadata_controller11
CMN700.cmn700_tag_cache. metadata_controller11.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller110	ExportTest.CMN700.cmn700_tag_cachemetadata_controller110
CMN700.cmn700_tag_cache. metadata_controller110.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller111	ExportTest.CMN700.cmn700_tag_cachemetadata_controller111
CMN700.cmn700_tag_cache. metadata_controller111.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller112	ExportTest.CMN700.cmn700_tag_cachemetadata_controller112
CMN700.cmn700_tag_cache. metadata_controller112.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller113	ExportTest.CMN700.cmn700_tag_cachemetadata_controller113
CMN700.cmn700_tag_cache. metadata_controller113.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller114	ExportTest.CMN700.cmn700_tag_cachemetadata_controller114
CMN700.cmn700_tag_cache. metadata_controller114.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller115	ExportTest.CMN700.cmn700_tag_cachemetadata_controller115
CMN700.cmn700_tag_cache. metadata_controller115.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller116	ExportTest.CMN700.cmn700_tag_cachemetadata_controller116
CMN700.cmn700_tag_cache. metadata_controller116.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller117	ExportTest.CMN700.cmn700_tag_cachemetadata_controller117
CMN700.cmn700_tag_cache. metadata_controller117.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller118	ExportTest.CMN700.cmn700_tag_cachemetadata_controller118
CMN700.cmn700_tag_cache. metadata_controller118.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller119	ExportTest.CMN700.cmn700_tag_cachemetadata_controller119
CMN700.cmn700_tag_cache. metadata_controller119.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller12	ExportTest.CMN700.cmn700_tag_cachemetadata_controller12
CMN700.cmn700_tag_cache. metadata_controller12.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller120	ExportTest.CMN700.cmn700_tag_cachemetadata_controller120
CMN700.cmn700_tag_cache. metadata_controller120.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller121	ExportTest.CMN700.cmn700_tag_cachemetadata_controller121
CMN700.cmn700_tag_cache. metadata_controller121.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller122	ExportTest.CMN700.cmn700_tag_cachemetadata_controller122
CMN700.cmn700_tag_cache. metadata_controller122.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller123	ExportTest.CMN700.cmn700_tag_cachemetadata_controller123
CMN700.cmn700_tag_cache. metadata_controller123.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller124	ExportTest.CMN700.cmn700_tag_cachemetadata_controller124
CMN700.cmn700_tag_cache. metadata_controller124.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller125	ExportTest.CMN700.cmn700_tag_cachemetadata_controller125
CMN700.cmn700_tag_cache. metadata_controller125.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller126	ExportTest.CMN700.cmn700_tag_cachemetadata_controller126
CMN700.cmn700_tag_cache. metadata_controller126.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller127	ExportTest.CMN700.cmn700_tag_cachemetadata_controller127
CMN700.cmn700_tag_cache. metadata_controller127.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller13	ExportTest.CMN700.cmn700_tag_cachemetadata_controller13
CMN700.cmn700_tag_cache. metadata_controller13.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller14	ExportTest.CMN700.cmn700_tag_cachemetadata_controller14
CMN700.cmn700_tag_cache. metadata_controller14.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller15	ExportTest.CMN700.cmn700_tag_cachemetadata_controller15

InstanceName	ComponentName
CMN700.cmn700_tag_cache. metadata_controller15.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller16	ExportTest.CMN700.cmn700_tag_cachemetadata_controller16
CMN700.cmn700_tag_cache. metadata_controller16.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller17	ExportTest.CMN700.cmn700_tag_cachemetadata_controller17
CMN700.cmn700_tag_cache. metadata_controller17.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller18	ExportTest.CMN700.cmn700_tag_cachemetadata_controller18
CMN700.cmn700_tag_cache. metadata_controller18.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller19	ExportTest.CMN700.cmn700_tag_cachemetadata_controller19
CMN700.cmn700_tag_cache. metadata_controller19.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller2	ExportTest.CMN700.cmn700_tag_cachemetadata_controller2
CMN700.cmn700_tag_cache. metadata_controller2.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller20	ExportTest.CMN700.cmn700_tag_cachemetadata_controller20
CMN700.cmn700_tag_cache. metadata_controller20.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller21	ExportTest.CMN700.cmn700_tag_cachemetadata_controller21
CMN700.cmn700_tag_cache. metadata_controller21.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller22	ExportTest.CMN700.cmn700_tag_cachemetadata_controller22
CMN700.cmn700_tag_cache. metadata_controller22.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller23	ExportTest.CMN700.cmn700_tag_cachemetadata_controller23
CMN700.cmn700_tag_cache. metadata_controller23.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller24	ExportTest.CMN700.cmn700_tag_cachemetadata_controller24
CMN700.cmn700_tag_cache. metadata_controller24.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller25	ExportTest.CMN700.cmn700_tag_cachemetadata_controller25
CMN700.cmn700_tag_cache. metadata_controller25.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller26	ExportTest.CMN700.cmn700_tag_cachemetadata_controller26
CMN700.cmn700_tag_cache. metadata_controller26.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller27	ExportTest.CMN700.cmn700_tag_cachemetadata_controller27
CMN700.cmn700_tag_cache. metadata_controller27.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller28	ExportTest.CMN700.cmn700_tag_cachemetadata_controller28
CMN700.cmn700_tag_cache. metadata_controller28.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller29	ExportTest.CMN700.cmn700_tag_cachemetadata_controller29
CMN700.cmn700_tag_cache. metadata_controller29.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller3	ExportTest.CMN700.cmn700_tag_cachemetadata_controller3
CMN700.cmn700_tag_cache. metadata_controller3.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller30	ExportTest.CMN700.cmn700_tag_cachemetadata_controller30
CMN700.cmn700_tag_cache. metadata_controller30.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller31	ExportTest.CMN700.cmn700_tag_cachemetadata_controller31
CMN700.cmn700_tag_cache. metadata_controller31.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller32	ExportTest.CMN700.cmn700_tag_cachemetadata_controller32
CMN700.cmn700_tag_cache. metadata_controller32.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller33	ExportTest.CMN700.cmn700_tag_cachemetadata_controller33
CMN700.cmn700_tag_cache. metadata_controller33.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller34	ExportTest.CMN700.cmn700_tag_cachemetadata_controller34
CMN700.cmn700_tag_cache. metadata_controller34.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller35	ExportTest.CMN700.cmn700_tag_cachemetadata_controller35
CMN700.cmn700_tag_cache. metadata_controller35.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller36	ExportTest.CMN700.cmn700_tag_cachemetadata_controller36
CMN700.cmn700_tag_cache. metadata_controller36.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller37	ExportTest.CMN700.cmn700_tag_cachemetadata_controller37
CMN700.cmn700_tag_cache. metadata_controller37.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller38	ExportTest.CMN700.cmn700_tag_cachemetadata_controller38
CMN700.cmn700_tag_cache. metadata_controller38.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller39	ExportTest.CMN700.cmn700_tag_cachemetadata_controller39
CMN700.cmn700_tag_cache. metadata_controller39.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller4	ExportTest.CMN700.cmn700_tag_cachemetadata_controller4
CMN700.cmn700_tag_cache. metadata_controller4.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller40	ExportTest.CMN700.cmn700_tag_cachemetadata_controller40
CMN700.cmn700_tag_cache. metadata_controller40.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller41	ExportTest.CMN700.cmn700_tag_cachemetadata_controller41

InstanceName	ComponentName
CMN700.cmn700_tag_cache. metadata_controller41.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller42	ExportTest.CMN700.cmn700_tag_cachemetadata_controller42
CMN700.cmn700_tag_cache. metadata_controller42.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller43	ExportTest.CMN700.cmn700_tag_cachemetadata_controller43
CMN700.cmn700_tag_cache. metadata_controller43.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller44	ExportTest.CMN700.cmn700_tag_cachemetadata_controller44
CMN700.cmn700_tag_cache. metadata_controller44.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller45	ExportTest.CMN700.cmn700_tag_cachemetadata_controller45
CMN700.cmn700_tag_cache. metadata_controller45.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller46	ExportTest.CMN700.cmn700_tag_cachemetadata_controller46
CMN700.cmn700_tag_cache. metadata_controller46.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller47	ExportTest.CMN700.cmn700_tag_cachemetadata_controller47
CMN700.cmn700_tag_cache. metadata_controller47.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller48	ExportTest.CMN700.cmn700_tag_cachemetadata_controller48
CMN700.cmn700_tag_cache. metadata_controller48.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller49	ExportTest.CMN700.cmn700_tag_cachemetadata_controller49
CMN700.cmn700_tag_cache. metadata_controller49.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller5	ExportTest.CMN700.cmn700_tag_cachemetadata_controller5
CMN700.cmn700_tag_cache. metadata_controller5.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller50	ExportTest.CMN700.cmn700_tag_cachemetadata_controller50
CMN700.cmn700_tag_cache. metadata_controller50.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller51	ExportTest.CMN700.cmn700_tag_cachemetadata_controller51
CMN700.cmn700_tag_cache. metadata_controller51.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller52	ExportTest.CMN700.cmn700_tag_cachemetadata_controller52
CMN700.cmn700_tag_cache. metadata_controller52.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller53	ExportTest.CMN700.cmn700_tag_cachemetadata_controller53
CMN700.cmn700_tag_cache. metadata_controller53.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller54	ExportTest.CMN700.cmn700_tag_cachemetadata_controller54
CMN700.cmn700_tag_cache. metadata_controller54.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller55	ExportTest.CMN700.cmn700_tag_cachemetadata_controller55
CMN700.cmn700_tag_cache. metadata_controller55.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller56	ExportTest.CMN700.cmn700_tag_cachemetadata_controller56
CMN700.cmn700_tag_cache. metadata_controller56.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller57	ExportTest.CMN700.cmn700_tag_cachemetadata_controller57
CMN700.cmn700_tag_cache. metadata_controller57.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller58	ExportTest.CMN700.cmn700_tag_cachemetadata_controller58
CMN700.cmn700_tag_cache. metadata_controller58.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller59	ExportTest.CMN700.cmn700_tag_cachemetadata_controller59
CMN700.cmn700_tag_cache. metadata_controller59.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller6	ExportTest.CMN700.cmn700_tag_cachemetadata_controller6
CMN700.cmn700_tag_cache. metadata_controller6.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller60	ExportTest.CMN700.cmn700_tag_cachemetadata_controller60
CMN700.cmn700_tag_cache. metadata_controller60.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller61	ExportTest.CMN700.cmn700_tag_cachemetadata_controller61
CMN700.cmn700_tag_cache. metadata_controller61.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller62	ExportTest.CMN700.cmn700_tag_cachemetadata_controller62
CMN700.cmn700_tag_cache. metadata_controller62.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller63	ExportTest.CMN700.cmn700_tag_cachemetadata_controller63
CMN700.cmn700_tag_cache. metadata_controller63.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller64	ExportTest.CMN700.cmn700_tag_cachemetadata_controller64
CMN700.cmn700_tag_cache. metadata_controller64.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller65	ExportTest.CMN700.cmn700_tag_cachemetadata_controller65
CMN700.cmn700_tag_cache. metadata_controller65.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller66	ExportTest.CMN700.cmn700_tag_cachemetadata_controller66
CMN700.cmn700_tag_cache. metadata_controller66.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller67	ExportTest.CMN700.cmn700_tag_cachemetadata_controller67
CMN700.cmn700_tag_cache. metadata_controller67.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller68	ExportTest.CMN700.cmn700_tag_cachemetadata_controller68

InstanceName	ComponentName
CMN700.cmn700_tag_cache. metadata_controller68.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller69	ExportTest.CMN700.cmn700_tag_cachemetadata_controller69
CMN700.cmn700_tag_cache. metadata_controller69.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller7	ExportTest.CMN700.cmn700_tag_cachemetadata_controller7
CMN700.cmn700_tag_cache. metadata_controller7.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller70	ExportTest.CMN700.cmn700_tag_cachemetadata_controller70
CMN700.cmn700_tag_cache. metadata_controller70.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller71	ExportTest.CMN700.cmn700_tag_cachemetadata_controller71
CMN700.cmn700_tag_cache. metadata_controller71.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller72	ExportTest.CMN700.cmn700_tag_cachemetadata_controller72
CMN700.cmn700_tag_cache. metadata_controller72.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller73	ExportTest.CMN700.cmn700_tag_cachemetadata_controller73
CMN700.cmn700_tag_cache. metadata_controller73.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller74	ExportTest.CMN700.cmn700_tag_cachemetadata_controller74
CMN700.cmn700_tag_cache. metadata_controller74.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller75	ExportTest.CMN700.cmn700_tag_cachemetadata_controller75
CMN700.cmn700_tag_cache. metadata_controller75.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller76	ExportTest.CMN700.cmn700_tag_cachemetadata_controller76
CMN700.cmn700_tag_cache. metadata_controller76.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller77	ExportTest.CMN700.cmn700_tag_cachemetadata_controller77
CMN700.cmn700_tag_cache. metadata_controller77.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller78	ExportTest.CMN700.cmn700_tag_cachemetadata_controller78
CMN700.cmn700_tag_cache. metadata_controller78.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller79	ExportTest.CMN700.cmn700_tag_cachemetadata_controller79
CMN700.cmn700_tag_cache. metadata_controller79.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller8	ExportTest.CMN700.cmn700_tag_cachemetadata_controller8
CMN700.cmn700_tag_cache. metadata_controller8.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller80	ExportTest.CMN700.cmn700_tag_cachemetadata_controller80
CMN700.cmn700_tag_cache. metadata_controller80.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller81	ExportTest.CMN700.cmn700_tag_cachemetadata_controller81
CMN700.cmn700_tag_cache. metadata_controller81.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller82	ExportTest.CMN700.cmn700_tag_cachemetadata_controller82
CMN700.cmn700_tag_cache. metadata_controller82.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller83	ExportTest.CMN700.cmn700_tag_cachemetadata_controller83
CMN700.cmn700_tag_cache. metadata_controller83.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller84	ExportTest.CMN700.cmn700_tag_cachemetadata_controller84
CMN700.cmn700_tag_cache. metadata_controller84.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller85	ExportTest.CMN700.cmn700_tag_cachemetadata_controller85
CMN700.cmn700_tag_cache. metadata_controller85.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller86	ExportTest.CMN700.cmn700_tag_cachemetadata_controller86
CMN700.cmn700_tag_cache. metadata_controller86.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller87	ExportTest.CMN700.cmn700_tag_cachemetadata_controller87
CMN700.cmn700_tag_cache. metadata_controller87.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller88	ExportTest.CMN700.cmn700_tag_cachemetadata_controller88
CMN700.cmn700_tag_cache. metadata_controller88.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller89	ExportTest.CMN700.cmn700_tag_cachemetadata_controller89
CMN700.cmn700_tag_cache. metadata_controller89.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller9	ExportTest.CMN700.cmn700_tag_cachemetadata_controller9
CMN700.cmn700_tag_cache. metadata_controller9.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller90	ExportTest.CMN700.cmn700_tag_cachemetadata_controller90
CMN700.cmn700_tag_cache. metadata_controller90.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller91	ExportTest.CMN700.cmn700_tag_cachemetadata_controller91
CMN700.cmn700_tag_cache. metadata_controller91.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller92	ExportTest.CMN700.cmn700_tag_cachemetadata_controller92
CMN700.cmn700_tag_cache. metadata_controller92.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller93	ExportTest.CMN700.cmn700_tag_cachemetadata_controller93
CMN700.cmn700_tag_cache. metadata_controller93.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller94	ExportTest.CMN700.cmn700_tag_cachemetadata_controller94

InstanceName	ComponentName
CMN700.cmn700_tag_cache. metadata_controller94.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller95	ExportTest.CMN700.cmn700_tag_cachemetadata_controller95
CMN700.cmn700_tag_cache. metadata_controller95.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller96	ExportTest.CMN700.cmn700_tag_cachemetadata_controller96
CMN700.cmn700_tag_cache. metadata_controller96.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller97	ExportTest.CMN700.cmn700_tag_cachemetadata_controller97
CMN700.cmn700_tag_cache. metadata_controller97.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller98	ExportTest.CMN700.cmn700_tag_cachemetadata_controller98
CMN700.cmn700_tag_cache. metadata_controller98.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller99	ExportTest.CMN700.cmn700_tag_cachemetadata_controller99
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
CMN700.cmn700_tag_cache.remapper116	PVBusMapper
CMN700.cmn700_tag_cache.remapper117	PVBusMapper
CMN700.cmn700_tag_cache.remapper118	PVBusMapper
CMN700.cmn700_tag_cache.remapper119	PVBusMapper

InstanceName	ComponentName
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
CMN700.cmn700_tag_cache.remapper37	PVBusMapper
CMN700.cmn700_tag_cache.remapper38	PVBusMapper
CMN700.cmn700_tag_cache.remapper39	PVBusMapper
CMN700.cmn700_tag_cache.remapper4	PVBusMapper

InstanceName	ComponentName
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
CMN700.cmn700_tag_cache.remapper72	PVBusMapper
CMN700.cmn700_tag_cache.remapper73	PVBusMapper
CMN700.cmn700_tag_cache.remapper74	PVBusMapper
CMN700.cmn700_tag_cache.remapper75	PVBusMapper

InstanceName	ComponentName
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.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

InstanceName	ComponentName
CMN700.snf_mapper	PVBusMapper

This model has the following MTI trace components:

Table 3-955: 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
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

InstanceName	ComponentName
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	ExportTest.CMN700.cmn700_tag_cachemetadata_controller0
CMN700.cmn700_tag_cache.metadata_controller0.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller1	ExportTest.CMN700.cmn700_tag_cachemetadata_controller1
CMN700.cmn700_tag_cache.metadata_controller1.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller10	ExportTest.CMN700.cmn700_tag_cachemetadata_controller10
CMN700.cmn700_tag_cache.metadata_controller10.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller100	ExportTest.CMN700.cmn700_tag_cachemetadata_controller100
CMN700.cmn700_tag_cache.metadata_controller100.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller101	ExportTest.CMN700.cmn700_tag_cachemetadata_controller101
CMN700.cmn700_tag_cache.metadata_controller101.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller102	ExportTest.CMN700.cmn700_tag_cachemetadata_controller102
CMN700.cmn700_tag_cache.metadata_controller102.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller103	ExportTest.CMN700.cmn700_tag_cachemetadata_controller103
CMN700.cmn700_tag_cache.metadata_controller103.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller104	ExportTest.CMN700.cmn700_tag_cachemetadata_controller104
CMN700.cmn700_tag_cache.metadata_controller104.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller105	ExportTest.CMN700.cmn700_tag_cachemetadata_controller105
CMN700.cmn700_tag_cache.metadata_controller105.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller106	ExportTest.CMN700.cmn700_tag_cachemetadata_controller106
CMN700.cmn700_tag_cache.metadata_controller106.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller107	ExportTest.CMN700.cmn700_tag_cachemetadata_controller107
CMN700.cmn700_tag_cache.metadata_controller107.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller108	ExportTest.CMN700.cmn700_tag_cachemetadata_controller108

InstanceName	ComponentName
CMN700.cmn700_tag_cache. metadata_controller108.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller109	ExportTest.CMN700.cmn700_tag_cachemetadata_controller109
CMN700.cmn700_tag_cache. metadata_controller109.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller11	ExportTest.CMN700.cmn700_tag_cachemetadata_controller11
CMN700.cmn700_tag_cache. metadata_controller11.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller110	ExportTest.CMN700.cmn700_tag_cachemetadata_controller110
CMN700.cmn700_tag_cache. metadata_controller110.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller111	ExportTest.CMN700.cmn700_tag_cachemetadata_controller111
CMN700.cmn700_tag_cache. metadata_controller111.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller112	ExportTest.CMN700.cmn700_tag_cachemetadata_controller112
CMN700.cmn700_tag_cache. metadata_controller112.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller113	ExportTest.CMN700.cmn700_tag_cachemetadata_controller113
CMN700.cmn700_tag_cache. metadata_controller113.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller114	ExportTest.CMN700.cmn700_tag_cachemetadata_controller114
CMN700.cmn700_tag_cache. metadata_controller114.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller115	ExportTest.CMN700.cmn700_tag_cachemetadata_controller115
CMN700.cmn700_tag_cache. metadata_controller115.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller116	ExportTest.CMN700.cmn700_tag_cachemetadata_controller116
CMN700.cmn700_tag_cache. metadata_controller116.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller117	ExportTest.CMN700.cmn700_tag_cachemetadata_controller117
CMN700.cmn700_tag_cache. metadata_controller117.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller118	ExportTest.CMN700.cmn700_tag_cachemetadata_controller118
CMN700.cmn700_tag_cache. metadata_controller118.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller119	ExportTest.CMN700.cmn700_tag_cachemetadata_controller119
CMN700.cmn700_tag_cache. metadata_controller119.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller12	ExportTest.CMN700.cmn700_tag_cachemetadata_controller12
CMN700.cmn700_tag_cache. metadata_controller12.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller120	ExportTest.CMN700.cmn700_tag_cachemetadata_controller120
CMN700.cmn700_tag_cache. metadata_controller120.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller121	ExportTest.CMN700.cmn700_tag_cachemetadata_controller121
CMN700.cmn700_tag_cache. metadata_controller121.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller122	ExportTest.CMN700.cmn700_tag_cachemetadata_controller122
CMN700.cmn700_tag_cache. metadata_controller122.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller123	ExportTest.CMN700.cmn700_tag_cachemetadata_controller123
CMN700.cmn700_tag_cache. metadata_controller123.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller124	ExportTest.CMN700.cmn700_tag_cachemetadata_controller124
CMN700.cmn700_tag_cache. metadata_controller124.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller125	ExportTest.CMN700.cmn700_tag_cachemetadata_controller125
CMN700.cmn700_tag_cache. metadata_controller125.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller126	ExportTest.CMN700.cmn700_tag_cachemetadata_controller126
CMN700.cmn700_tag_cache. metadata_controller126.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller127	ExportTest.CMN700.cmn700_tag_cachemetadata_controller127
CMN700.cmn700_tag_cache. metadata_controller127.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller13	ExportTest.CMN700.cmn700_tag_cachemetadata_controller13
CMN700.cmn700_tag_cache. metadata_controller13.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller14	ExportTest.CMN700.cmn700_tag_cachemetadata_controller14
CMN700.cmn700_tag_cache. metadata_controller14.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller15	ExportTest.CMN700.cmn700_tag_cachemetadata_controller15
CMN700.cmn700_tag_cache. metadata_controller15.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller16	ExportTest.CMN700.cmn700_tag_cachemetadata_controller16
CMN700.cmn700_tag_cache. metadata_controller16.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller17	ExportTest.CMN700.cmn700_tag_cachemetadata_controller17
CMN700.cmn700_tag_cache. metadata_controller17.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller18	ExportTest.CMN700.cmn700_tag_cachemetadata_controller18
CMN700.cmn700_tag_cache. metadata_controller18.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller19	ExportTest.CMN700.cmn700_tag_cachemetadata_controller19
CMN700.cmn700_tag_cache. metadata_controller19.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller2	ExportTest.CMN700.cmn700_tag_cachemetadata_controller2

InstanceName	ComponentName
CMN700.cmn700_tag_cache. metadata_controller2.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller20	ExportTest.CMN700.cmn700_tag_cachemetadata_controller20
CMN700.cmn700_tag_cache. metadata_controller20.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller21	ExportTest.CMN700.cmn700_tag_cachemetadata_controller21
CMN700.cmn700_tag_cache. metadata_controller21.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller22	ExportTest.CMN700.cmn700_tag_cachemetadata_controller22
CMN700.cmn700_tag_cache. metadata_controller22.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller23	ExportTest.CMN700.cmn700_tag_cachemetadata_controller23
CMN700.cmn700_tag_cache. metadata_controller23.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller24	ExportTest.CMN700.cmn700_tag_cachemetadata_controller24
CMN700.cmn700_tag_cache. metadata_controller24.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller25	ExportTest.CMN700.cmn700_tag_cachemetadata_controller25
CMN700.cmn700_tag_cache. metadata_controller25.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller26	ExportTest.CMN700.cmn700_tag_cachemetadata_controller26
CMN700.cmn700_tag_cache. metadata_controller26.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller27	ExportTest.CMN700.cmn700_tag_cachemetadata_controller27
CMN700.cmn700_tag_cache. metadata_controller27.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller28	ExportTest.CMN700.cmn700_tag_cachemetadata_controller28
CMN700.cmn700_tag_cache. metadata_controller28.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller29	ExportTest.CMN700.cmn700_tag_cachemetadata_controller29
CMN700.cmn700_tag_cache. metadata_controller29.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller3	ExportTest.CMN700.cmn700_tag_cachemetadata_controller3
CMN700.cmn700_tag_cache. metadata_controller3.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller30	ExportTest.CMN700.cmn700_tag_cachemetadata_controller30
CMN700.cmn700_tag_cache. metadata_controller30.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller31	ExportTest.CMN700.cmn700_tag_cachemetadata_controller31
CMN700.cmn700_tag_cache. metadata_controller31.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller32	ExportTest.CMN700.cmn700_tag_cachemetadata_controller32
CMN700.cmn700_tag_cache. metadata_controller32.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller33	ExportTest.CMN700.cmn700_tag_cachemetadata_controller33
CMN700.cmn700_tag_cache. metadata_controller33.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller34	ExportTest.CMN700.cmn700_tag_cachemetadata_controller34
CMN700.cmn700_tag_cache. metadata_controller34.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller35	ExportTest.CMN700.cmn700_tag_cachemetadata_controller35
CMN700.cmn700_tag_cache. metadata_controller35.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller36	ExportTest.CMN700.cmn700_tag_cachemetadata_controller36
CMN700.cmn700_tag_cache. metadata_controller36.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller37	ExportTest.CMN700.cmn700_tag_cachemetadata_controller37
CMN700.cmn700_tag_cache. metadata_controller37.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller38	ExportTest.CMN700.cmn700_tag_cachemetadata_controller38
CMN700.cmn700_tag_cache. metadata_controller38.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller39	ExportTest.CMN700.cmn700_tag_cachemetadata_controller39
CMN700.cmn700_tag_cache. metadata_controller39.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller4	ExportTest.CMN700.cmn700_tag_cachemetadata_controller4
CMN700.cmn700_tag_cache. metadata_controller4.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller40	ExportTest.CMN700.cmn700_tag_cachemetadata_controller40
CMN700.cmn700_tag_cache. metadata_controller40.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller41	ExportTest.CMN700.cmn700_tag_cachemetadata_controller41
CMN700.cmn700_tag_cache. metadata_controller41.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller42	ExportTest.CMN700.cmn700_tag_cachemetadata_controller42
CMN700.cmn700_tag_cache. metadata_controller42.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller43	ExportTest.CMN700.cmn700_tag_cachemetadata_controller43
CMN700.cmn700_tag_cache. metadata_controller43.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller44	ExportTest.CMN700.cmn700_tag_cachemetadata_controller44
CMN700.cmn700_tag_cache. metadata_controller44.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller45	ExportTest.CMN700.cmn700_tag_cachemetadata_controller45
CMN700.cmn700_tag_cache. metadata_controller45.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller46	ExportTest.CMN700.cmn700_tag_cachemetadata_controller46

InstanceName	ComponentName
CMN700.cmn700_tag_cache. metadata_controller46.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller47	ExportTest.CMN700.cmn700_tag_cachemetadata_controller47
CMN700.cmn700_tag_cache. metadata_controller47.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller48	ExportTest.CMN700.cmn700_tag_cachemetadata_controller48
CMN700.cmn700_tag_cache. metadata_controller48.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller49	ExportTest.CMN700.cmn700_tag_cachemetadata_controller49
CMN700.cmn700_tag_cache. metadata_controller49.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller5	ExportTest.CMN700.cmn700_tag_cachemetadata_controller5
CMN700.cmn700_tag_cache. metadata_controller5.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller50	ExportTest.CMN700.cmn700_tag_cachemetadata_controller50
CMN700.cmn700_tag_cache. metadata_controller50.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller51	ExportTest.CMN700.cmn700_tag_cachemetadata_controller51
CMN700.cmn700_tag_cache. metadata_controller51.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller52	ExportTest.CMN700.cmn700_tag_cachemetadata_controller52
CMN700.cmn700_tag_cache. metadata_controller52.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller53	ExportTest.CMN700.cmn700_tag_cachemetadata_controller53
CMN700.cmn700_tag_cache. metadata_controller53.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller54	ExportTest.CMN700.cmn700_tag_cachemetadata_controller54
CMN700.cmn700_tag_cache. metadata_controller54.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller55	ExportTest.CMN700.cmn700_tag_cachemetadata_controller55
CMN700.cmn700_tag_cache. metadata_controller55.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller56	ExportTest.CMN700.cmn700_tag_cachemetadata_controller56
CMN700.cmn700_tag_cache. metadata_controller56.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller57	ExportTest.CMN700.cmn700_tag_cachemetadata_controller57
CMN700.cmn700_tag_cache. metadata_controller57.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller58	ExportTest.CMN700.cmn700_tag_cachemetadata_controller58
CMN700.cmn700_tag_cache. metadata_controller58.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller59	ExportTest.CMN700.cmn700_tag_cachemetadata_controller59
CMN700.cmn700_tag_cache. metadata_controller59.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller6	ExportTest.CMN700.cmn700_tag_cachemetadata_controller6
CMN700.cmn700_tag_cache. metadata_controller6.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller60	ExportTest.CMN700.cmn700_tag_cachemetadata_controller60
CMN700.cmn700_tag_cache. metadata_controller60.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller61	ExportTest.CMN700.cmn700_tag_cachemetadata_controller61
CMN700.cmn700_tag_cache. metadata_controller61.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller62	ExportTest.CMN700.cmn700_tag_cachemetadata_controller62
CMN700.cmn700_tag_cache. metadata_controller62.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller63	ExportTest.CMN700.cmn700_tag_cachemetadata_controller63
CMN700.cmn700_tag_cache. metadata_controller63.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller64	ExportTest.CMN700.cmn700_tag_cachemetadata_controller64
CMN700.cmn700_tag_cache. metadata_controller64.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller65	ExportTest.CMN700.cmn700_tag_cachemetadata_controller65
CMN700.cmn700_tag_cache. metadata_controller65.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller66	ExportTest.CMN700.cmn700_tag_cachemetadata_controller66
CMN700.cmn700_tag_cache. metadata_controller66.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller67	ExportTest.CMN700.cmn700_tag_cachemetadata_controller67
CMN700.cmn700_tag_cache. metadata_controller67.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller68	ExportTest.CMN700.cmn700_tag_cachemetadata_controller68
CMN700.cmn700_tag_cache. metadata_controller68.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller69	ExportTest.CMN700.cmn700_tag_cachemetadata_controller69
CMN700.cmn700_tag_cache. metadata_controller69.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller7	ExportTest.CMN700.cmn700_tag_cachemetadata_controller7
CMN700.cmn700_tag_cache. metadata_controller7.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller70	ExportTest.CMN700.cmn700_tag_cachemetadata_controller70
CMN700.cmn700_tag_cache. metadata_controller70.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller71	ExportTest.CMN700.cmn700_tag_cachemetadata_controller71
CMN700.cmn700_tag_cache. metadata_controller71.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller72	ExportTest.CMN700.cmn700_tag_cachemetadata_controller72

InstanceName	ComponentName
CMN700.cmn700_tag_cache. metadata_controller72.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller73	ExportTest.CMN700.cmn700_tag_cachemetadata_controller73
CMN700.cmn700_tag_cache. metadata_controller73.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller74	ExportTest.CMN700.cmn700_tag_cachemetadata_controller74
CMN700.cmn700_tag_cache. metadata_controller74.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller75	ExportTest.CMN700.cmn700_tag_cachemetadata_controller75
CMN700.cmn700_tag_cache. metadata_controller75.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller76	ExportTest.CMN700.cmn700_tag_cachemetadata_controller76
CMN700.cmn700_tag_cache. metadata_controller76.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller77	ExportTest.CMN700.cmn700_tag_cachemetadata_controller77
CMN700.cmn700_tag_cache. metadata_controller77.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller78	ExportTest.CMN700.cmn700_tag_cachemetadata_controller78
CMN700.cmn700_tag_cache. metadata_controller78.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller79	ExportTest.CMN700.cmn700_tag_cachemetadata_controller79
CMN700.cmn700_tag_cache. metadata_controller79.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller8	ExportTest.CMN700.cmn700_tag_cachemetadata_controller8
CMN700.cmn700_tag_cache. metadata_controller8.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller80	ExportTest.CMN700.cmn700_tag_cachemetadata_controller80
CMN700.cmn700_tag_cache. metadata_controller80.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller81	ExportTest.CMN700.cmn700_tag_cachemetadata_controller81
CMN700.cmn700_tag_cache. metadata_controller81.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller82	ExportTest.CMN700.cmn700_tag_cachemetadata_controller82
CMN700.cmn700_tag_cache. metadata_controller82.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller83	ExportTest.CMN700.cmn700_tag_cachemetadata_controller83
CMN700.cmn700_tag_cache. metadata_controller83.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller84	ExportTest.CMN700.cmn700_tag_cachemetadata_controller84
CMN700.cmn700_tag_cache. metadata_controller84.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller85	ExportTest.CMN700.cmn700_tag_cachemetadata_controller85
CMN700.cmn700_tag_cache. metadata_controller85.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller86	ExportTest.CMN700.cmn700_tag_cachemetadata_controller86
CMN700.cmn700_tag_cache. metadata_controller86.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller87	ExportTest.CMN700.cmn700_tag_cachemetadata_controller87
CMN700.cmn700_tag_cache. metadata_controller87.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller88	ExportTest.CMN700.cmn700_tag_cachemetadata_controller88
CMN700.cmn700_tag_cache. metadata_controller88.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller89	ExportTest.CMN700.cmn700_tag_cachemetadata_controller89
CMN700.cmn700_tag_cache. metadata_controller89.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller9	ExportTest.CMN700.cmn700_tag_cachemetadata_controller9
CMN700.cmn700_tag_cache. metadata_controller9.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller90	ExportTest.CMN700.cmn700_tag_cachemetadata_controller90
CMN700.cmn700_tag_cache. metadata_controller90.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller91	ExportTest.CMN700.cmn700_tag_cachemetadata_controller91
CMN700.cmn700_tag_cache. metadata_controller91.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller92	ExportTest.CMN700.cmn700_tag_cachemetadata_controller92
CMN700.cmn700_tag_cache. metadata_controller92.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller93	ExportTest.CMN700.cmn700_tag_cachemetadata_controller93
CMN700.cmn700_tag_cache. metadata_controller93.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller94	ExportTest.CMN700.cmn700_tag_cachemetadata_controller94
CMN700.cmn700_tag_cache. metadata_controller94.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller95	ExportTest.CMN700.cmn700_tag_cachemetadata_controller95
CMN700.cmn700_tag_cache. metadata_controller95.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller96	ExportTest.CMN700.cmn700_tag_cachemetadata_controller96
CMN700.cmn700_tag_cache. metadata_controller96.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller97	ExportTest.CMN700.cmn700_tag_cachemetadata_controller97
CMN700.cmn700_tag_cache. metadata_controller97.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller98	ExportTest.CMN700.cmn700_tag_cachemetadata_controller98
CMN700.cmn700_tag_cache. metadata_controller98.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller99	ExportTest.CMN700.cmn700_tag_cachemetadata_controller99

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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.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

CMN700 contains the following CADI targets:

- CMN700
- CMN_TAG_CACHE

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 `pvbus_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, `pvbus_s_rni[0-2]` maps to `RND0`, `pvbus_s_rni[3-5]` maps to `RND1` and `pvbus_s_rni[6-8]` maps to `RNI0`.

Similarly, SN-F and SBSX nodes are mapped to `pvbus_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 `pvbus_m_cxs[]` ports. The CXRH nodes, if any, are connected to `pvbus_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 `INTREQPPU` 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.

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 all 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 0b00000. 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 `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.
- Multiple CPA groups 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.
 - There is no support for AXU.
- 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 mxp_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 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.
- 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.

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.

Ports for CMN700

Table 3-956: 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
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_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

""

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"



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

Model the cache state.

`cmn700_tag_cache.metadata_controller.init_value`

Type

int

Default value

0xd

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

""

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

""

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

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

""

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 ≥ 64 bytes and a power of 2, defaulting to 4KiB. The maximum block size supported is 4KiB. The carveout region size must be ≥ 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

""

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

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. THIS PARAMETER HAS NO FUNCTIONALITY AT THE MOMENT.

cmn700_tag_cache.metadata_controller.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.

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

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

enable_a4s**Type**

bool

Default value

0x0

Enables A4S ports for GIC multi-chip routing.

enable_logger**Type**

bool

Default value

0x0

Enable PVBusLoggers for the downstream ports in the CMN model.

enable_ras**Type**

bool

Default value

0x0

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

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

""

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

Print information about the CCIX configuration.

print_cmn_config**Type**

bool

Default value

0x0

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

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

Does the top-level YML file describe node-addresses ?.

3.10.19 DMA350

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-957: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for DMA350

This model has the following Iris instances:

Table 3-958: DMA350 Iris instances

InstanceName	ComponentName
DMA350	DMA350
DMA350.pvbus_m0_bus_master	PVBusMaster
DMA350.pvbus_m1_bus_master	PVBusMaster

InstanceName	ComponentName
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
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 3-959: DMA350 MTI instances

InstanceName	ComponentName
DMA350	DMA350
DMA350.pvbus_m0_bus_master	PVBusMaster

InstanceName	ComponentName
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
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

DMA350 contains the following CADI targets:

- DMA350

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

Ports for DMA350

Table 3-960: 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
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

Name	Protocol	Type	Description
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
pvbust_m0	PVBus	Master	AXI5 Master 0 Interface
pvbust_m1	PVBus	Master	AXI5 Master 1 Interface
pvbust_s	PVBus	Slave	APB4 Slave Interface
pvbust_stream_in[16]	PVBus	Slave	AXI-Stream In Interface
pvbust_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

Address width of the bus interface.

AXI5_M1_ADDRESS_RANGES

Type

string

Default value

""

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

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

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

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_10_FIFO_DEPTH**Type**

int

Default value

0x2

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_11_FIFO_DEPTH**Type**

int

Default value

0x2

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_12_FIFO_DEPTH

Type

int

Default value

0x2

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_13_FIFO_DEPTH

Type

int

Default value

0x2

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_14_FIFO_DEPTH

Type

int

Default value

0x2

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_15_FIFO_DEPTH

Type

int

Default value

0x2

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

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

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

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

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

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

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

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_8_FIFO_DEPTH

Type

int

Default value

0x2

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_9_FIFO_DEPTH

Type

int

Default value

0x2

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

int

Default value

0x1

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

int

Default value

0x1

.

CH_GPO_MASK

Type

int

Default value

0x0

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**

int

Default value

0x1

.

CH_STREAM_MASK**Type**

int

Default value

0x0

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

Data width of the bus interface.

DISABLE_DEVICE**Type**

bool

Default value

0x0

Disable device and ignore all interfaces.

DUMP_CONFIG**Type**

bool

Default value

0x0

Display DMA-350 DMAC parameters.

GPO_WIDTH**Type**

int

Default value

0x1

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

Number of configurable DMA channels.

NUM_TRIGGER_IN**Type**

int

Default value

0x2

Number of trigger input ports.

NUM_TRIGGER_OUT**Type**

int

Default value

0x2

Number of trigger output ports.

SECEXT_PRESENT**Type**

bool

Default value

0x1

Enables Trustzone security support.

3.10.20 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 3-961: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for DMC500

This model has the following Iris instances:

Table 3-962: DMC500 Iris instances

InstanceName	ComponentName
DMC500	DMC500
DMC500.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-963: DMC500 MTI instances

InstanceName	ComponentName
DMC500	DMC-500
DMC500.busslave	PVBusSlave

DMC500 contains the following CADI targets:

- DMC500

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.

Ports for DMC500

Table 3-964: 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

Default Region Secure attributes. Only bits 31,30 set Secure RD/WR enable.

default_region_id_access

Type

int

Default value

0x0

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

Always allow debug access to memory.

3.10.21 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 3-965: IP revisions support

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for DMC520

This model has the following Iris instances:

Table 3-966: DMC520 Iris instances

InstanceName	ComponentName
DMC520	DMC520
DMC520.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-967: DMC520 MTI instances

InstanceName	ComponentName
DMC520	DMC-520
DMC520.busslave	PVBusSlave

DMC520 contains the following CADI targets:

- DMC520

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_pvbus_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.

Ports for DMC520

Table 3-968: Ports

Name	Protocol	Type	Description
<code>all_or_interrupt_signal</code>	Signal	Master	A combined interrupt that is the logical OR of the other interrupts.
<code>apb_pvbus_s</code>	PVBus	Slave	Programmers interface to program and control the DMC-520.
<code>arch_fsm_interrupt_signal</code>	Signal	Master	The DMC has detected a change in the architectural state.
<code>failed_access_interrupt_signal</code>	Signal	Master	The DMC has detected a system request that has failed a permissions check and a previously detected assertion was not cleared.
<code>filter_pvbus_m</code>	PVBus	Master	DMC master port to memory.
<code>filter_pvbus_s</code>	PVBus	Slave	System interface.
<code>reset_signal</code>	Signal	Slave	DMC reset.

Name	Protocol	Type	Description
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

Override default block-all behavior of DMC. Allow access to memory.

passthrough_debug_access

Type

bool

Default value

0x0

Always allow debug access to memory.

3.10.22 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 3-969: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for DMC620

This model has the following Iris instances:

Table 3-970: DMC620 Iris instances

InstanceName	ComponentName
DMC620	DMC620
DMC620.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-971: DMC620 MTI instances

InstanceName	ComponentName
DMC620	DMC-620
DMC620.busslave	PVBusSlave

DMC620 contains the following CADI targets:

- DMC620

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_pvbus_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.

Ports for DMC620

Table 3-972: 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.

Name	Protocol	Type	Description
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

Override default block-all behavior of DMC. Allow access to memory.

passthrough_debug_access

Type

bool

Default value

0x0

Always allow debug access to memory.

3.10.23 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 3-973: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for DMC_400

This model has the following Iris instances:

Table 3-974: DMC_400 Iris instances

InstanceName	ComponentName
DMC_400	DMC_400
DMC_400.apb_slave	PVBusSlave

InstanceName	ComponentName
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 3-975: 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

DMC_400 contains the following CADI targets:

- DMC_400

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.

Ports for DMC_400

Table 3-976: 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

Does the controller support ECC?.

IF_CHIP0**Type**

int

Default value

0xffffffffffffffff

Set this parameter to 0 if memory is connected.

IF_CHIP1**Type**

int

Default value

0xffffffffffffffff

Set this parameter to 0 if memory is connected.

IF_CHIP2**Type**

int

Default value

0xffffffffffffffff

Set this parameter to 0 if memory is connected.

IF_CHIP3**Type**

int

Default value

0xffffffffffffffff

Set this parameter to 0 if memory is connected.

MEMORY_WIDTH**Type**

int

Default value

0x20

Valid widths are 16, 32 or 64 bits.

diagnostics

Type
int
Default value
0x0

Diagnostics.

revision_string

Type
string
Default value
"r0p1"

Revision.

3.10.24 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 3-977: IP revisions support

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for EthosU55

This model has the following Iris instances:

Table 3-978: 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 3-979: EthosU55 MTI instances

InstanceName	ComponentName
EthosU55.pvbusmaster0	PVBusMaster
EthosU55.pvbusmaster1	PVBusMaster
EthosU55.pvbuslave	PVBusSlave

EthosU55 contains the following CADI targets:

- EthosU55

About EthosU55

The EthosU55 model has the following parameters:

diagnostics

Enables additional information messages from the EthosU55 component about the NPU configuration and the PORPL/PORSL port values. Use any nonzero integer to enable all messages. The default value of 0 disables them.

extra_args

Reserved for future use. We recommend using this parameter only with customized instructions from Arm Technical Support.

num_macs

Controls the number of 8x8 MACs performed per cycle. The valid values are 32, 64, 128, or 256. The default value is 128.



- The EthosU55 model does not expose its registers through CADI or Iris.
- The resetn_in signal is active-LOW.

Ports for EthosU55

Table 3-980: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	NPU clock signal
irq_out	Signal	Master	Sends interrupt requests to the external host application processor
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

Parameters for EthosU55

diagnostics

Type
int
Default value
0x0

Enable diagnostic messages.

extra_args

Type
string
Default value
""

Reserved for future use. We recommend using this parameter only with customized instructions from Arm Technical Support (support-esl@arm.com).

num_macs

Type
int
Default value
0x80

Number of 8x8 MACs performed per cycle.

3.10.25 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 3-981: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for EthosU65

This model has the following Iris instances:

Table 3-982: 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 3-983: EthosU65 MTI instances

InstanceName	ComponentName
EthosU65.pvbusmaster0	PVBusMaster
EthosU65.pvbusmaster1	PVBusMaster
EthosU65.pvbusslave	PVBusSlave

EthosU65 contains the following CADI targets:

- EthosU65

About EthosU65

The EthosU65 model has the following parameters:

diagnostics

Enables additional information messages from the EthosU65 component about the NPU configuration and the PORPL/PORSL port values. Use any nonzero integer to enable all messages. The default value of 0 disables them.

extra_args

Reserved for future use. We recommend using this parameter only with customized instructions from Arm Technical Support.

num_macs

Controls the number of 8x8 MACs performed per cycle. The valid values are 256 and 512. The default value is 256.



Note

- The EthosU65 model does not expose its registers through CADI or Iris.
- The resetn_in signal is active-LOW.

Ports for EthosU65

Table 3-984: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	NPU clock signal
irq_out	Signal	Master	Sends interrupt requests to the external host application processor
popl_in	Signal	Slave	The Power-On-Reset Privilege Level (PORPL). LOW means User level. HIGH means Privileged level

Name	Protocol	Type	Description
posl_in	Signal	Slave	The Power-On-Reset Security Level (PORSL). LOW means Secure. HIGH means Non-secure.
pvbush_m0	PVBus	Master	Port 0 for NPU to access external memory
pvbush_m1	PVBus	Master	Port 1 for NPU to access external memory
pvbush_s	PVBus	Slave	Port to access NPU control registers
reseth_in	Signal	Slave	NPU reset signal

Parameters for EthosU65

diagnostics

Type

int

Default value

0x0

Enable diagnostic messages.

extra_args

Type

string

Default value

""

Reserved for future use. We recommend using this parameter only with customized instructions from Arm Technical Support.

num_macs

Type

int

Default value

0x100

Number of 8x8 MACs performed per cycle.

3.10.26 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 3-985: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Changes in 11.27.17

Table 3-986: Model quality changes

From	To
pre	rel

Iris and MTI instances for EthosU85

This model has the following Iris instances:

Table 3-987: 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 3-988: EthosU85 MTI instances

InstanceName	ComponentName
EthosU85.pvbusmaster0	PVBusMaster
EthosU85.pvbusmaster1	PVBusMaster
EthosU85.pvbusslave	PVBusSlave

EthosU85 contains the following CADI targets:

- EthosU85

Ports for EthosU85

Table 3-989: Ports

Name	Protocol	Type	Description
cfgextcap_in	Value	Slave	The configuration of capabilities for DRAM AXI ports (32 bits). Sampled with soft and hard reset.
cfgexthash0_in	Value_64	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	Value	Slave	The configuration of capabilities for SRAM AXI ports (32 bits). Sampled with soft and hard reset.

Name	Protocol	Type	Description
cfgsramhash0_in	Value_64	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	Value_64	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	ClockSignal	Slave	NPU clock signal
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
resetsn_in	Signal	Slave	NPU reset signal (active-LOW)

Parameters for EthosU85

diagnostics

Type

int

Default value

0x0

Enable diagnostic messages.

extra_args

Type

string

Default value

""

Reserved for future use. We recommend using this parameter only with customized instructions from Arm Technical Support.

num_macs

Type

int

Default value

0x80

Number of 8x8 MACs performed per cycle.

3.10.27 Firewall

Firewall IP. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-990: IP revisions support

Revision	Quality level
rOp0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for Firewall

This model has the following Iris instances:

Table 3-991: 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 3-992: 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

Firewall contains the following CADI targets:

- Firewall

Ports for Firewall

Table 3-993: 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

Default value

[illegible]

Maximum address field width of the Bus interfaces of the Firewall Component. 0x00 - 0x04 - Reserved 0x05 - 5 bits of address 0x06 - 6 bits of address ... 0x40 - 64 bits of address All other values are Reserved. When a Firewall Component supports more than one Bus Slave or Master interface this field is set to the largest width used. .

BYPASS IF ST

Type

int

Default value

0x1

Status of the bypass interface implemented by the Firewall. 0x0 - Bypass interface disabled. 0x1 - Bypass interface enabled. All other values are Reserved.

BYPASS VLD

Type

int

Default value

0x1

Validity for the bypass mode implemented by the Firewall. 0x0 - Bypass mode invalid implemented. 0x1 - Bypass mode valid implemented. All other values are Reserved.

CONFIG FILE

Type

string

Default value

■■■■

Firewall Configuration file.

INST SPT

Type

string

Minimum Region Size. This field is Reserved and treated as RAZ/WI when PE.0 is implemented.
0x0 - 32B, 0x1 - 64B, 0x2 - 128B, 0x3 - 256B, 0x4 - 512B, 0x5 - 1KB, 0x6 - 2KB, 0x7 - 4KB.
Another value will get the default one (0) .

MST_ID WIDTH

Type

string

Default value

[illegible]

Maximum MasterID field width of the Bus interfaces of the Firewall Component. 0x00 - 1 bit of MasterID 0x01 - 2 bits of MasterID ... 0x1F - 32 bits of MasterID When a Firewall Component supports more than one Bus Slave or Master interface this field is set to the largest width used .

MXRS

Type

string

Default value

[illegible]

The Maximum Region Size the Firewall Component Supports 0x05 - 32B. 0x06 - 64B. ... 0x0C - 4KB. 0x0D - 8KB. ... 0x40 - 16EB. All other values are Reserved. This field is Reserved and treated as RAZ/WI invalid for a Firewall Component which implements PE.0 .

NUM_FC

Type

int

Default value

0x1f

Number of Firewall Components implemented. The number of Firewall Components which are implemented in the Firewall is NUM_FC+1. For Firewall Components, other than 0, this field is Reserved and treated as RAZ/WI. .

NUM MPE

Type

string

Default value

[illegible]

Number of MPEs implemented, per region, in the Firewall Component. 0b00 - 1 MPE per region, 0b01 - 2 MPE per region, 0b10 - 3 MPE per region, 0b11 - 4 MPE per region, This field is Reserved and treated as RAZ/WI when PE.0 is implemented. .

NUM_RGN

Type

string

Default value

[illegible]

Number of regions implemented in the Firewall Component. The number of regions implemented is NUM_RGN+1. This field is Reserved and treated as RAZ/WI when PE.0 is implemented.

PE LVL

Type

string

Default value

[illegible]

Level of the Protection Extension implemented by the Firewall Component. 0x0 - PE.0 is implemented. 0x1 - PE.1 is implemented. 0x2 - PE.2 is implemented. All other values are Reserved. .

PRIV SPT

Type

string

Default value

[illegible]

Firewall Component support for checking the privileged level of the incoming transaction and setting the privileged level of the outgoing transaction. Ob0 - Not supported, Ob1 - Supported .

PROT SIZE

Type

string

Default value

[illegible]

Protection size, as defined by the Protection Size interface. The value of this field indicates the range of addresses which the Firewall Component protects 0x00 - 0B. 0x05 - 32B. 0x06 - 64B. ... 0x07 - 4KB. 0x08 - 8KB. ... 0x40 - 16EB. All other values are Reserved and treated as if 0x00. This field is Reserved and treated as RAZ/WI invalid for a Firewall Component which implements PE.0. When this field reads as 0x00 all transactions are considered to fail the protection logic checks. .

RSE LVL

Type

string

TE_LVL

Type

string

Default value

[illegible]

Level of the Translation Extension implemented by Firewall Component 0. 0x0 - TE.0 is implemented. 0x1 - TE.1 is implemented. 0x2 - TE.2 is implemented. All other values are Reserved. (This field is always 0x0 when the FC_CAPO.PE_LVL is 0x0.) .

3.10.28 GIC500

GLC500 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 3-994: IP revisions support

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for GIC500

This model has the following Iris instances:

Table 3-995: 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 3-996: GIC500 MTI instances

InstanceName	ComponentName
GIC500	GICv3IRI
GIC500.ITS0	GICv3InterruptTranslationService
GIC500.rd_0	GICv3RedistributorInternal
GIC500.rd_0_0	GICv3RedistributorInternal

InstanceName	ComponentName
GIC500.rd_0_0_0	GICv3RedistributorInternal
GIC500.rd_0_0_0_0	GICv3Redistributor
GIC500.rd_t1	GICv3Distributor

GIC500 contains the following CADI targets:

- GIC500

About GIC500

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.

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:

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`.

Ports for GIC500

Table 3-998: 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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.
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.

Name	Protocol	Type	Description
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.
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

If true, the GICD_ITARGETS registers are RAZ/WI.

ITS-count

Type

int

Default value

0x1

Number of Interrupt Translation Services to be instantiated (0=none).

ITS-device-bits

Type

int

Default value

0x10

Number of bits supported for ITS device IDs.

ITS-threaded-command-queue

Type

bool

Default value

0x1

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

SPI-count

Type

int

Default value

0xe0

Number of SPIs that are implemented.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**Type**

bool

Default value

0x0

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

Number of cores within cluster 0.

cpus_per_cluster_1**Type**

int

Default value

0x1

Number of cores within cluster 1.

cpus_per_cluster_10**Type**

int

Default value

0x1

Number of cores within cluster 10.

cpus_per_cluster_11**Type**

int

Default value

0x1

Number of cores within cluster 11.

cpus_per_cluster_12**Type**

int

Default value

0x1

Number of cores within cluster 12.

cpus_per_cluster_13**Type**

int

Default value

0x1

Number of cores within cluster 13.

cpus_per_cluster_14**Type**

int

Default value

0x1

Number of cores within cluster 14.

cpus_per_cluster_15**Type**

int

Default value

0x1

Number of cores within cluster 15.

cpus_per_cluster_16**Type**

int

Default value

0x1

Number of cores within cluster 16.

cpus_per_cluster_17**Type**

int

Default value

0x1

Number of cores within cluster 17.

cpus_per_cluster_18**Type**

int

Default value

0x1

Number of cores within cluster 18.

cpus_per_cluster_19**Type**

int

Default value

0x1

Number of cores within cluster 19.

cpus_per_cluster_2**Type**

int

Default value

0x1

Number of cores within cluster 2.

cpus_per_cluster_20**Type**

int

Default value

0x1

Number of cores within cluster 20.

cpus_per_cluster_21**Type**

int

Default value

0x1

Number of cores within cluster 21.

cpus_per_cluster_22**Type**

int

Default value

0x1

Number of cores within cluster 22.

cpus_per_cluster_23**Type**

int

Default value

0x1

Number of cores within cluster 23.

cpus_per_cluster_24**Type**

int

Default value

0x1

Number of cores within cluster 24.

cpus_per_cluster_25**Type**

int

Default value

0x1

Number of cores within cluster 25.

cpus_per_cluster_26**Type**

int

Default value

0x1

Number of cores within cluster 26.

cpus_per_cluster_27**Type**

int

Default value

0x1

Number of cores within cluster 27.

cpus_per_cluster_28**Type**

int

Default value

0x1

Number of cores within cluster 28.

cpus_per_cluster_29**Type**

int

Default value

0x1

Number of cores within cluster 29.

cpus_per_cluster_3**Type**

int

Default value

0x1

Number of cores within cluster 3.

cpus_per_cluster_30**Type**

int

Default value

0x1

Number of cores within cluster 30.

cpus_per_cluster_31**Type**

int

Default value

0x1

Number of cores within cluster 31.

cpus_per_cluster_4**Type**

int

Default value

0x1

Number of cores within cluster 4.

cpus_per_cluster_5**Type**

int

Default value

0x1

Number of cores within cluster 5.

cpus_per_cluster_6**Type**

int

Default value

0x1

Number of cores within cluster 6.

cpus_per_cluster_7**Type**

int

Default value

0x1

Number of cores within cluster 7.

cpus_per_cluster_8**Type**

int

Default value

0x1

Number of cores within cluster 8.

cpus_per_cluster_9**Type**

int

Default value

0x1

Number of cores within cluster 9.

delay-ITS-accesses**Type**

bool

Default value

0x1

Delay accesses from the ITS until GICR_SYNCRR is read.

delay-redistributor-accesses**Type**

bool

Default value

0x1

Delay memory accesses from the redistributor until GICR_SYNCRR is read.

enable_protocol_checking**Type**

bool

Default value

0x0

Enable/disable protocol checking at cpu interface.

enabled**Type**

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

has-two-security-states**Type**

bool

Default value

0x1

If true, has two security states.

num_clusters**Type**

int

Default value

0x1

Number of implemented affinity level1 clusters.

print-memory-map**Type**

bool

Default value

0x0

Print memory map to stdout.

redistributor-threaded-sync**Type**

bool

Default value

0x1

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

reg-base**Type**

int

Default value

0x10000000

GIC500 base address.

using-generated-memorymap**Type**

bool

Default value

0x1

Use the generated memorymap for the GIC500 and warn if superfluous parameters are passed.

wakeup-on-reset**Type**

bool

Default value

0x0

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.

3.10.29 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 3-999: 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 23.

Iris and MTI instances for GIC500_ClusterPorts

This model has the following Iris instances:

Table 3-1000: 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 3-1001: GIC500_ClusterPorts MTI instances

InstanceName	ComponentName
GIC500_ClusterPorts	GICv3IRI
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

GIC500_ClusterPorts contains the following CADI targets:

- GIC500

About GIC500_ClusterPorts

An alternate version of GIC500, identical to the above except for the CPUIF ports being exposed as an array per cluster

Ports for GIC500_ClusterPorts

Table 3-1002: Ports

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	Disable some SPIs signal.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.
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_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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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

If true, the GICD_ITARGETS registers are RAZ/WI.

ITS-count

Type

int

Default value

0x1

Number of Interrupt Translation Services to be instantiated (0=none).

ITS-device-bits**Type**

int

Default value

0x10

Number of bits supported for ITS device IDs.

ITS-threaded-command-queue**Type**

bool

Default value

0x1

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

SPI-count**Type**

int

Default value

0xe0

Number of SPIs that are implemented.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**Type**

bool

Default value

0x0

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

Number of cores within cluster 0.

cpus_per_cluster_1**Type**

int

Default value

0x1

Number of cores within cluster 1.

cpus_per_cluster_10**Type**

int

Default value

0x1

Number of cores within cluster 10.

cpus_per_cluster_11**Type**

int

Default value

0x1

Number of cores within cluster 11.

cpus_per_cluster_12**Type**

int

Default value

0x1

Number of cores within cluster 12.

cpus_per_cluster_13**Type**

int

Default value

0x1

Number of cores within cluster 13.

cpus_per_cluster_14**Type**

int

Default value

0x1

Number of cores within cluster 14.

cpus_per_cluster_15**Type**

int

Default value

0x1

Number of cores within cluster 15.

cpus_per_cluster_16**Type**

int

Default value

0x1

Number of cores within cluster 16.

cpus_per_cluster_17**Type**

int

Default value

0x1

Number of cores within cluster 17.

cpus_per_cluster_18**Type**

int

Default value

0x1

Number of cores within cluster 18.

cpus_per_cluster_19**Type**

int

Default value

0x1

Number of cores within cluster 19.

cpus_per_cluster_2**Type**

int

Default value

0x1

Number of cores within cluster 2.

cpus_per_cluster_20**Type**

int

Default value

0x1

Number of cores within cluster 20.

cpus_per_cluster_21**Type**

int

Default value

0x1

Number of cores within cluster 21.

cpus_per_cluster_22**Type**

int

Default value

0x1

Number of cores within cluster 22.

cpus_per_cluster_23**Type**

int

Default value

0x1

Number of cores within cluster 23.

cpus_per_cluster_24**Type**

int

Default value

0x1

Number of cores within cluster 24.

cpus_per_cluster_25**Type**

int

Default value

0x1

Number of cores within cluster 25.

cpus_per_cluster_26**Type**

int

Default value

0x1

Number of cores within cluster 26.

cpus_per_cluster_27**Type**

int

Default value

0x1

Number of cores within cluster 27.

cpus_per_cluster_28**Type**

int

Default value

0x1

Number of cores within cluster 28.

cpus_per_cluster_29**Type**

int

Default value

0x1

Number of cores within cluster 29.

cpus_per_cluster_3**Type**

int

Default value

0x1

Number of cores within cluster 3.

cpus_per_cluster_30**Type**

int

Default value

0x1

Number of cores within cluster 30.

cpus_per_cluster_31**Type**

int

Default value

0x1

Number of cores within cluster 31.

cpus_per_cluster_4**Type**

int

Default value

0x1

Number of cores within cluster 4.

cpus_per_cluster_5**Type**

int

Default value

0x1

Number of cores within cluster 5.

cpus_per_cluster_6**Type**

int

Default value

0x1

Number of cores within cluster 6.

cpus_per_cluster_7**Type**

int

Default value

0x1

Number of cores within cluster 7.

cpus_per_cluster_8**Type**

int

Default value

0x1

Number of cores within cluster 8.

cpus_per_cluster_9**Type**

int

Default value

0x1

Number of cores within cluster 9.

delay-ITS-accesses**Type**

bool

Default value

0x1

Delay accesses from the ITS until GICR_SYNCRR is read.

delay-redistributor-accesses**Type**

bool

Default value

0x1

Delay memory accesses from the redistributor until GICR_SYNCRR is read.

enable_protocol_checking**Type**

bool

Default value

0x0

Enable/disable protocol checking at cpu interface.

enabled**Type**

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

has-two-security-states**Type**

bool

Default value

0x1

If true, has two security states.

num_clusters**Type**

int

Default value

0x1

Number of implemented affinity level1 clusters.

print-memory-map**Type**

bool

Default value

0x0

Print memory map to stdout.

redistributor-threaded-sync**Type**

bool

Default value

0x1

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

reg-base**Type**

int

Default value

0x10000000

GIC500 base address.

using-generated-memorymap**Type**

bool

Default value

0x1

Use the generated memorymap for the GIC500 and warn if superfluous parameters are passed.

wakeup-on-reset**Type**

bool

Default value

0x0

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.

3.10.30 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 3-1003: 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 23.

Iris and MTI instances for GIC500_Filter

This model has the following Iris instances:

Table 3-1004: 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 3-1005: GIC500_Filter MTI instances

InstanceName	ComponentName
GIC500_Filter	GICv3IRI
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

GIC500_Filter contains the following CADI targets:

- GIC500

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:

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`.

Ports for GIC500_Filter

Table 3-1007: 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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.
pvbuse_filtermiss_m	PVBus	Master	passthrough for transactions not targetting one of the pages associated with the IRI.
pvbuse_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[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_Filter

GICD_ITARGETSR-RAZWI

Type

bool

Default value

0x0

If true, the GICD_ITARGETS registers are RAZ/WI.

ITS-count

Type

int

Default value

0x1

Number of Interrupt Translation Services to be instantiated (0=none).

ITS-device-bits

Type

int

Default value

0x10

Number of bits supported for ITS device IDs.

ITS-threaded-command-queue

Type

bool

Default value

0x1

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

SPI-count

Type

int

Default value

0xe0

Number of SPIs that are implemented.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**Type**

bool

Default value

0x0

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

Number of cores within cluster 0.

cpus_per_cluster_1**Type**

int

Default value

0x1

Number of cores within cluster 1.

cpus_per_cluster_10**Type**

int

Default value

0x1

Number of cores within cluster 10.

cpus_per_cluster_11**Type**

int

Default value

0x1

Number of cores within cluster 11.

cpus_per_cluster_12**Type**

int

Default value

0x1

Number of cores within cluster 12.

cpus_per_cluster_13**Type**

int

Default value

0x1

Number of cores within cluster 13.

cpus_per_cluster_14**Type**

int

Default value

0x1

Number of cores within cluster 14.

cpus_per_cluster_15**Type**

int

Default value

0x1

Number of cores within cluster 15.

cpus_per_cluster_16**Type**

int

Default value

0x1

Number of cores within cluster 16.

cpus_per_cluster_17**Type**

int

Default value

0x1

Number of cores within cluster 17.

cpus_per_cluster_18**Type**

int

Default value

0x1

Number of cores within cluster 18.

cpus_per_cluster_19**Type**

int

Default value

0x1

Number of cores within cluster 19.

cpus_per_cluster_2**Type**

int

Default value

0x1

Number of cores within cluster 2.

cpus_per_cluster_20**Type**

int

Default value

0x1

Number of cores within cluster 20.

cpus_per_cluster_21**Type**

int

Default value

0x1

Number of cores within cluster 21.

cpus_per_cluster_22**Type**

int

Default value

0x1

Number of cores within cluster 22.

cpus_per_cluster_23**Type**

int

Default value

0x1

Number of cores within cluster 23.

cpus_per_cluster_24**Type**

int

Default value

0x1

Number of cores within cluster 24.

cpus_per_cluster_25**Type**

int

Default value

0x1

Number of cores within cluster 25.

cpus_per_cluster_26**Type**

int

Default value

0x1

Number of cores within cluster 26.

cpus_per_cluster_27**Type**

int

Default value

0x1

Number of cores within cluster 27.

cpus_per_cluster_28**Type**

int

Default value

0x1

Number of cores within cluster 28.

cpus_per_cluster_29**Type**

int

Default value

0x1

Number of cores within cluster 29.

cpus_per_cluster_3**Type**

int

Default value

0x1

Number of cores within cluster 3.

cpus_per_cluster_30**Type**

int

Default value

0x1

Number of cores within cluster 30.

cpus_per_cluster_31**Type**

int

Default value

0x1

Number of cores within cluster 31.

cpus_per_cluster_4**Type**

int

Default value

0x1

Number of cores within cluster 4.

cpus_per_cluster_5**Type**

int

Default value

0x1

Number of cores within cluster 5.

cpus_per_cluster_6**Type**

int

Default value

0x1

Number of cores within cluster 6.

cpus_per_cluster_7**Type**

int

Default value

0x1

Number of cores within cluster 7.

cpus_per_cluster_8**Type**

int

Default value

0x1

Number of cores within cluster 8.

cpus_per_cluster_9**Type**

int

Default value

0x1

Number of cores within cluster 9.

delay-ITS-accesses**Type**

bool

Default value

0x1

Delay accesses from the ITS until GICR_SYNCRR is read.

delay-redistributor-accesses**Type**

bool

Default value

0x1

Delay memory accesses from the redistributor until GICR_SYNCRR is read.

enable_protocol_checking**Type**

bool

Default value

0x0

Enable/disable protocol checking at cpu interface.

enabled**Type**

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

has-two-security-states**Type**

bool

Default value

0x1

If true, has two security states.

num_clusters**Type**

int

Default value

0x1

Number of implemented affinity level1 clusters.

print-memory-map**Type**

bool

Default value

0x0

Print memory map to stdout.

redistributor-threaded-sync**Type**

bool

Default value

0x1

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

reg-base**Type**

int

Default value

0x10000000

GIC500 base address.

using-generated-memorymap**Type**

bool

Default value

0x1

Use the generated memorymap for the GIC500 and warn if superfluous parameters are passed.

wakeup-on-reset**Type**

bool

Default value

0x0

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.

3.10.31 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 3-1008: IP revisions support

Revision	Quality level
r1p6	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Changes in 11.27.17

Parameters added:

- `gicp-allow-ns-reset`

Iris and MTI instances for GIC600

This model has the following Iris instances:

Table 3-1009: 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 3-1010: GIC600 MTI instances

InstanceName	ComponentName
GIC600	GICv3IRI
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

GIC600 contains the following CADI targets:

- GIC600

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 `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 GIC600_Filter variant has a `pvbuss_filtermiss_m` port, to which any transactions on the `pvbuss_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:

PVBus attribute	Bits used	Property encoded	Notes
MasterID	31:0	Master ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



The pvbus_m port does not use MasterID, ExtendedID, Or UserFlags.

Ports for GIC600

Table 3-1012: 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	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 GIC600

CPU-affinities

Type

string

Default value

0.0.0.0

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

""

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

0x0

GICD_IIDR and GICR_IIDR value. When the parameter is not given or is set to zero, the GIC600 model is configured to have the latest revision.

ITS-ID-bits**Type**

int

Default value

0x10

Number of interrupt bits supported by ITS.

ITS-collection-ID-bits**Type**

int

Default value

0x8

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

ITS-count**Type**

int

Default value

0x1

Number of Interrupt Translation Services to be instantiated (0=none).

ITS-device-bits**Type**

int

Default value

0x10

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

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

RAS-FI-support**Type**

bool

Default value

0x0

If true, fault handling interrupt is supported. Not supported otherwise.

RAS-UE-support**Type**

bool

Default value

0x0

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

RAS-UI-support**Type**

bool

Default value

0x0

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

SPI-blocks**Type**

int

Default value

0x1e

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

affinity-width

Type

string

Default value

"4.8.8.8"

A dotted quad indicating the bitwidth of fields at each affinity level.

chip-id

Type

int

Default value

0x0

Chip ID when multichip operation is enabled.

chip-select-affinity-level

Type

int

Default value

0x3

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

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

direct-lpi-support

Type

bool

Default value

0x0

Enable support for LPI operations through GICR registers.

enable-multichip-operation**Type**

bool

Default value

0x0

Enables multi-chip operation between Distributors in distributed GIC IRI.

enabled**Type**

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

gicp-allow-ns-reset**Type**

bool

Default value

0x1

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

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

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

max-pe-on-chip**Type**

int

Default value

0x4

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

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

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

GIC600 dedistributor power management is done by updating GICR_PWRR register.

reg-base**Type**

int

Default value

0x2c010000

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.

3.10.32 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 3-1013: IP revisions support

Revision	Quality level
r0p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Changes in 11.27.17

Parameters added:

- `gicp-allow-ns-reset`

Iris and MTI instances for GIC600AE

This model has the following Iris instances:

Table 3-1014: GIC600AE Iris instances

InstanceName	ComponentName
GIC600AE	GIC_IRI
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_t1	GICv3Distributor

This model has the following MTI trace components:

Table 3-1015: GIC600AE MTI instances

InstanceName	ComponentName
GIC600AE	GICv3IRI
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_t1	GICv3Distributor

GIC600AE contains the following CADI targets:

- GIC600AE

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 `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 GIC600AE_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 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.



- 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:

PVBus attribute	Bits used	Property encoded	Notes
MasterID	31:0	Master ID that invoked the register access	–

PVBus attribute	Bits used	Property encoded	Notes
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



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.

Ports for GIC600AE

Table 3-1017: 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.
fmu_error_int	Signal	Master	FuSa FMU error interrupt signal
fmu_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.

Name	Protocol	Type	Description
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.
ppi_in_138[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 138.

Name	Protocol	Type	Description
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.
ppi_in_173[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 173.

Name	Protocol	Type	Description
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.
ppi_in_208[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 208.

Name	Protocol	Type	Description
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.
ppi_in_243[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 243.

Name	Protocol	Type	Description
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.
ppi_in_49[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 49.

Name	Protocol	Type	Description
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.
ppi_in_84[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 84.

Name	Protocol	Type	Description
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 GIC600AE

CPU-affinities

Type

string

Default value

0.0.0.0

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

""

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRL. 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

0x0

GICD_IIDR and GICR_IIDR value. When the parameter is not given or is set to zero, the GIC600 model is configured to have the latest revision.

ITS-ID-bits

Type

int

Default value

0x10

Number of interrupt bits supported by ITS.

ITS-collection-ID-bits

Type

int

Default value

0x8

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

ITS-count**Type**

int

Default value

0x1

Number of Interrupt Translation Services to be instantiated (0=none).

ITS-device-bits**Type**

int

Default value

0x10

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

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

RAS-FI-support**Type**

bool

Default value

0x0

If true, fault handling interrupt is supported. Not supported otherwise.

RAS-UE-support**Type**

bool

Default value

0x0

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

RAS-UI-support**Type**

bool

Default value

0x0

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

SPI-blocks**Type**

int

Default value

0x1e

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

affinity-width**Type**

string

Default value

"4.8.8.8"

A dotted quad indicating the bitwidth of fields at each affinity level.

chip-id**Type**

int

Default value

0x0

Chip ID when multichip operation is enabled.

chip-select-affinity-level**Type**

int

Default value

0x3

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

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

direct-lpi-support**Type**

bool

Default value

0x0

Enable support for LPI operations through GICR registers.

enable-multichip-operation**Type**

bool

Default value

0x0

Enables multi-chip operation between Distributors in distributed GIC IRI.

enabled**Type**

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

gicp-allow-ns-reset**Type**

bool

Default value

0x1

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

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

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

max-pe-on-chip**Type**

int

Default value

0x4

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

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

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

GIC600 dedistributor power management is done by updating GICR_PWRR register.

reg-base

Type
int

Default value
0x2c010000

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.

3.10.33 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 3-1018: 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 23.

Changes in 11.27.17

Parameters added:

- gicp-allow-ns-reset

Iris and MTI instances for GIC600AE_Filter

This model has the following Iris instances:

Table 3-1019: GIC600AE_Filter Iris instances

InstanceName	ComponentName
GIC600AE_Filter	GIC_IRI
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 3-1020: GIC600AE_Filter MTI instances

InstanceName	ComponentName
GIC600AE_Filter	GICv3IRI
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

GIC600AE_Filter contains the following CADI targets:

- GIC600AE

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 `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 `GIC600AE_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 `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:

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.

Ports for GIC600AE_Filter

Table 3-1022: 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.
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_filtermiss_m	PVBus	Master	Memory bus out. Transactions not filtered by the component.

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	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

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

""

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

0x0

GICD_IIDR and GICR_IIDR value. When the parameter is not given or is set to zero, the GIC600 model is configured to have the latest revision.

ITS-ID-bits**Type**

int

Default value

0x10

Number of interrupt bits supported by ITS.

ITS-collection-ID-bits**Type**

int

Default value

0x8

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

ITS-count**Type**

int

Default value

0x1

Number of Interrupt Translation Services to be instantiated (0=none).

ITS-device-bits**Type**

int

Default value

0x10

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

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

RAS-FI-support**Type**

bool

Default value

0x0

If true, fault handling interrupt is supported. Not supported otherwise.

RAS-UE-support**Type**

bool

Default value

0x0

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

RAS-UI-support**Type**

bool

Default value

0x0

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

SPI-blocks**Type**

int

Default value

0x1e

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

affinity-width**Type**

string

Default value

"4.8.8.8"

A dotted quad indicating the bitwidth of fields at each affinity level.

chip-id**Type**

int

Default value

0x0

Chip ID when multichip operation is enabled.

chip-select-affinity-level**Type**

int

Default value

0x3

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

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

direct-lpi-support**Type**

bool

Default value

0x0

Enable support for LPI operations through GICR registers.

enable-multichip-operation**Type**

bool

Default value

0x0

Enables multi-chip operation between Distributors in distributed GIC IRI.

enabled**Type**

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

gicp-allow-ns-reset**Type**

bool

Default value

0x1

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

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

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

max-pe-on-chip**Type**

int

Default value

0x4

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

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

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

GIC600 redistributor power management is done by updating GICR_PWRR register.

reg-base

Type

int

Default value

0x2c010000

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.

3.10.34 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 3-1023: 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 23.

Changes in 11.27.17

Parameters added:

- `gicp-allow-ns-reset`

Iris and MTI instances for GIC600_Filter

This model has the following Iris instances:

Table 3-1024: 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 3-1025: GIC600_Filter MTI instances

InstanceName	ComponentName
GIC600_Filter	GICv3IRI
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_t1	GICv3Distributor

GIC600_Filter contains the following CADI targets:

- GIC600

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:

PVBus attribute	Bits used	Property encoded	Notes
MasterID	31:0	Master ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



The `pvbus_m` port does not use `MasterID`, `ExtendedID`, or `UserFlags`.

Ports for GIC600_Filter

Table 3-1027: Ports

Name	Protocol	Type	Description
<code>chip_id</code>	Value	Slave	<code>chip_id</code> port which is valid from GIC600 r1p2. Writing to this port for prior GIC600 version will be ignored.
<code>cpu_active_s[256]</code>	Signal	Slave	CPUActive pins.
<code>po_reset</code>	Signal	Slave	Reset.
<code>ppi_in_0[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.
<code>ppi_in_1[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
<code>ppi_in_10[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 10.
<code>ppi_in_100[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 100.
<code>ppi_in_101[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 101.
<code>ppi_in_102[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 102.
<code>ppi_in_103[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 103.
<code>ppi_in_104[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 104.
<code>ppi_in_105[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 105.
<code>ppi_in_106[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 106.
<code>ppi_in_107[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 107.
<code>ppi_in_108[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 108.
<code>ppi_in_109[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 109.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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

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

""

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

0x0

GICD_IIDR and GICR_IIDR value. When the parameter is not given or is set to zero, the GIC600 model is configured to have the latest revision.

ITS-ID-bits**Type**

int

Default value

0x10

Number of interrupt bits supported by ITS.

ITS-collection-ID-bits**Type**

int

Default value

0x8

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

ITS-count**Type**

int

Default value

0x1

Number of Interrupt Translation Services to be instantiated (0=none).

ITS-device-bits**Type**

int

Default value

0x10

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

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

RAS-FI-support**Type**

bool

Default value

0x0

If true, fault handling interrupt is supported. Not supported otherwise.

RAS-UE-support**Type**

bool

Default value

0x0

If true, In-band uncorrected error reporing is supported. Not supported otherwise.

RAS-UI-support**Type**

bool

Default value

0x0

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

SPI-blocks**Type**

int

Default value

0x1e

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

affinity-width**Type**

string

Default value

"4.8.8.8"

A dotted quad indicating the bitwidth of fields at each affinity level.

chip-id**Type**

int

Default value

0x0

Chip ID when multichip operation is enabled.

chip-select-affinity-level**Type**

int

Default value

0x3

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

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

direct-lpi-support

Type

bool

Default value

0x0

Enable support for LPI operations through GICR registers.

enable-multichip-operation

Type

bool

Default value

0x0

Enables multi-chip operation between Distributors in distributed GIC IRI.

enabled

Type

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

gicp-allow-ns-reset

Type

bool

Default value

0x1

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

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

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

max-pe-on-chip**Type**

int

Default value

0x4

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

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

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

GIC600 redistributor power management is done by updating GICR_PWRR register.

reg-base**Type**

int

Default value

0x2c010000

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 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 3-1028: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Changes in 11.27.17

Parameters added:

- gicp-allow-ns-reset
- has-two-security-states

Parameters removed:

- DS-behaviour

Iris and MTI instances for GIC625

This model has the following Iris instances:

Table 3-1029: GIC625 Iris instances

InstanceName	ComponentName
GIC625	GIC_IRI
GIC625.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC625.rd_0	GICv3RedistributorInternal

InstanceName	ComponentName
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 3-1030: GIC625 MTI instances

InstanceName	ComponentName
GIC625	GICv3IRI
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

GIC625 contains the following CADI targets:

- GIC625

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.
 - 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.

Ports for GIC625

Table 3-1031: 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.
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

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

""

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

0x0

GICD_IIDR and GICR_IIDR value. When the parameter is not given or is set to zero, the GIC600 model is configured to have the latest revision.

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

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

RAS-FI-support**Type**

bool

Default value

0x0

If true, fault handling interrupt is supported. Not supported otherwise.

RAS-UE-support**Type**

bool

Default value

0x0

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

RAS-UI-support**Type**

bool

Default value

0x0

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

SPI-blocks**Type**

int

Default value

0x1e

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

affinity-width**Type**

string

Default value

"4.8.8.8"

A dotted quad indicating the bitwidth of fields at each affinity level.

chip-select-affinity-level**Type**

int

Default value

0x3

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

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

enabled**Type**

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

gicp-allow-ns-reset**Type**

bool

Default value

0x1

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

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

If true, has two security states.

max-cores-supported-by-GCI**Type**

int

Default value

0x8

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

print-memory-map**Type**

bool

Default value

0x0

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

GIC600 dedistributor power management is done by updating GICR_PWRR register.

reg-base

Type
int
Default value
0x2c010000

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.

3.10.36 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 3-1032: 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 23.

Changes in 11.27.17

Parameters added:

- `gicp-allow-ns-reset`
- `has-two-security-states`

Parameters removed:

- `DS-behaviour`

Iris and MTI instances for GIC625_Filter

This model has the following Iris instances:

Table 3-1033: 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 3-1034: GIC625_Filter MTI instances

InstanceName	ComponentName
GIC625_Filter	GICv3IRI
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

GIC625_Filter contains the following CADI targets:

- GIC625

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.



- Extended SPI is not supported.
- All implementation-specific registers and functionality are implemented.

Ports for GIC625_Filter

Table 3-1035: 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

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

""

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

0x0

GICD_IIDR and GICR_IIDR value. When the parameter is not given or is set to zero, the GIC600 model is configured to have the latest revision.

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

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

RAS-FI-support**Type**

bool

Default value

0x0

If true, fault handling interrupt is supported. Not supported otherwise.

RAS-UE-support**Type**

bool

Default value

0x0

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

RAS-UI-support**Type**

bool

Default value

0x0

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

SPI-blocks**Type**

int

Default value

0x1e

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

affinity-width**Type**

string

Default value

"4.8.8.8"

A dotted quad indicating the bitwidth of fields at each affinity level.

chip-select-affinity-level**Type**

int

Default value

0x3

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

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

enabled**Type**

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

gicp-allow-ns-reset**Type**

bool

Default value

0x1

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

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

If true, has two security states.

max-cores-supported-by-GCI**Type**

int

Default value

0x8

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

print-memory-map**Type**

bool

Default value

0x0

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

GIC600 dedistributor power management is done by updating GICR_PWRR register.

reg-base

Type

int

Default value

0x2c010000

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.

3.10.37 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 3-1036: IP revisions support

Revision	Quality level
r0p0	Full support
r1p0	Full support
r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Changes in 11.27.17

Parameters removed:

- `direct-lpi-support`
- `extended-spi-count`

Iris and MTI instances for GIC700

This model has the following Iris instances:

Table 3-1037: 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 3-1038: GIC700 MTI instances

InstanceName	ComponentName
GIC700	GICv3IRI
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_tl	GICv3Distributor

GIC700 contains the following CADI targets:

- GIC700

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` port for register accesses and a `pvbuss_m` port for the LPI-related traffic from redistributors and the ITS.

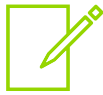
In addition, the GIC700_Filter variant has a `pvbuss_filtermiss_m` port, to which any transactions on the `pvbuss` port and not directed to a 4K page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC700 variant.

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:

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`.

Ports for GIC700

Table 3-1040: 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

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

""

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

GICD_CTLR-DS-1-means-secure-only**Type**

bool

Default value

0x0

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

GICD_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

IIDR**Type**

int

Default value

0x0

GICD_IIDR and GICR_IIDR value.

ITS-ID-bits**Type**

int

Default value

0x10

Number of interrupt bits supported by ITS.

ITS-collection-ID-bits**Type**

int

Default value

0x8

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0.

ITS-count**Type**

int

Default value

0x1

Number of Interrupt Translation Services to be instantiated (0=none).

ITS-device-bits**Type**

int

Default value

0x10

Number of bits supported for ITS device IDs.

ITS-enable-itt-address-verification**Type**

bool

Default value

0x0

If true, a transaction will be sent to ITT Address for verification.

ITS-hardware-collection-count**Type**

int

Default value

0x0

Number of hardware collections held exclusively in the ITS.

ITS-shared-vPE-table**Type**

int

Default value

0x0

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

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

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

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"

A dotted quad indicating the bitwidth of fields at each affinity level.

allow-LPIEN-clear**Type**

bool

Default value

0x1

Allow RW behaviour on GICR_CTLR.LPIEN instead of set once.

chip-count**Type**

int

Default value

0x10

The total number of chips supported.

chip-id**Type**

int

Default value

0x0

Chip ID when multichip operation is enabled.

chip-select-affinity-level**Type**

int

Default value

0x3

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

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

common-vPE-table-affinity**Type**

string

Default value

""

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

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

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

Enable GICv3 functionality; when false the component is inactive.

extended-ppi-count

Type

int

Default value

0x40

Number of extended PPI supported.

has-gicv4.1

Type

bool

Default value

0x1

Enable GICv4.1 functionality; when false the component is inactive.

has-two-security-states

Type

bool

Default value

0x1

If true, has two security states.

max-cores-supported-by-GCI**Type**

int

Default value

0x8

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

max-pe-on-chip**Type**

int

Default value

0x4

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

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"

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

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

GIC-700 dedistributor power management is done by updating GICR_PWRR register.

reg-base

Type

int

Default value

0x2c010000

GIC-700 base address.

3.10.38 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 3-1041: 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 23.

Changes in 11.27.17

Parameters removed:

- `direct-lpi-support`
- `extended-spi-count`

Iris and MTI instances for GIC700_Filter

This model has the following Iris instances:

Table 3-1042: 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 3-1043: GIC700_Filter MTI instances

InstanceName	ComponentName
GIC700_Filter	GICv3IRI
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

GIC700_Filter contains the following CADI targets:

- GIC700

About GIC700_Filter

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 `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 4K 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:

PVBus attribute	Bits used	Property encoded	Notes
MasterID	31:0	Master ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



The `pvbus_m` port does not use `MasterID`, `ExtendedID`, or `UserFlags`.

Ports for GIC700_Filter

Table 3-1045: Ports

Name	Protocol	Type	Description
<code>axi_stream_msi_s[32]</code>	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.
<code>chip_id</code>	Value	Slave	<code>chip_id</code> port for multichip operation
<code>cpu_active_s[256]</code>	Signal	Slave	CPUActive pins.
<code>extended_ppi_in_0[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.
<code>extended_ppi_in_1[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.
<code>extended_ppi_in_10[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 10.
<code>extended_ppi_in_100[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100
<code>extended_ppi_in_101[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101
<code>extended_ppi_in_102[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102
<code>extended_ppi_in_103[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103
<code>extended_ppi_in_104[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104
<code>extended_ppi_in_105[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105
<code>extended_ppi_in_106[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106
<code>extended_ppi_in_107[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107
<code>extended_ppi_in_108[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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
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.

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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.
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

Name	Protocol	Type	Description
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.
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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.
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
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_filtermiss_m	PVBus	Master	Memory bus out. Transactions not filtered by the component.
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	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

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

""

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

GICD_CTLR-DS-1-means-secure-only**Type**

bool

Default value

0x0

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

GICD_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

IIDR**Type**

int

Default value

0x0

GICD_IIDR and GICR_IIDR value.

ITS-ID-bits**Type**

int

Default value

0x10

Number of interrupt bits supported by ITS.

ITS-collection-ID-bits**Type**

int

Default value

0x8

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

ITS-count**Type**

int

Default value

0x1

Number of Interrupt Translation Services to be instantiated (0=none).

ITS-device-bits**Type**

int

Default value

0x10

Number of bits supported for ITS device IDs.

ITS-enable-itt-address-verification**Type**

bool

Default value

0x0

If true, a transaction will be sent to ITT Address for verification.

ITS-hardware-collection-count**Type**

int

Default value

0x0

Number of hardware collections held exclusively in the ITS.

ITS-shared-vPE-table

Type
int
Default value
0x0

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

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

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

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"

A dotted quad indicating the bitwidth of fields at each affinity level.

allow-LPIEN-clear**Type**

bool

Default value

0x1

Allow RW behaviour on `GICR_CTLR.LPIEN` instead of set once.

chip-count**Type**

int

Default value

0x10

The total number of chips supported.

chip-id**Type**

int

Default value

0x0

Chip ID when multichip operation is enabled.

chip-select-affinity-level**Type**

int

Default value

0x3

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

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

common-vPE-table-affinity**Type**

string

Default value

""

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

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

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

Enable GICv3 functionality; when false the component is inactive.

extended-ppi-count**Type**

int

Default value

0x40

Number of extended PPI supported.

has-gicv4.1**Type**

bool

Default value

0x1

Enable GICv4.1 functionality; when false the component is inactive.

has-two-security-states**Type**

bool

Default value

0x1

If true, has two security states.

max-cores-supported-by-GCI**Type**

int

Default value

0x8

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

max-pe-on-chip**Type**

int

Default value

0x4

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

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"

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

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

GIC-700 dedistributor power management is done by updating GICR_PWRR register.

reg-base

Type
int

Default value
0x2c010000

GIC-700 base address.

3.10.39 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 3-1046: 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 23.

Changes in 11.27.17

Parameters removed:

- `direct-lpi-support`
- `extended-spi-count`

Iris and MTI instances for GIC720AE

This model has the following Iris instances:

Table 3-1047: GIC720AE Iris instances

InstanceName	ComponentName
GIC720AE	GIC_IRI
GIC720AE.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC720AE.ITS0	GICv4InterruptTranslationService
GIC720AE.ITS0.bus_subordinate	PVBusSlave
GIC720AE.fmu	ExportTest.GIC720AEfmu
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 3-1048: GIC720AE MTI instances

InstanceName	ComponentName
GIC720AE	GICv3IRI
GIC720AE.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC720AE.ITS0	GICv4InterruptTranslationService
GIC720AE.ITS0.bus_subordinate	PVBusSlave
GIC720AE.fmu	ExportTest.GIC720AEfmu
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

GIC720AE contains the following CADI targets:

- GIC720AE

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](#).

Ports for GIC720AE

Table 3-1049: 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.
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

Name	Protocol	Type	Description
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
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.

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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.
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

Name	Protocol	Type	Description
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.
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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	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

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
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

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

""

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

GICD_CTLR-DS-1-means-secure-only**Type**

bool

Default value

0x0

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

GICD_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

IIDR**Type**

int

Default value

0x0

GICD_IIDR and GICR_IIDR value.

ITS-ID-bits**Type**

int

Default value

0x10

Number of interrupt bits supported by ITS.

ITS-collection-ID-bits**Type**

int

Default value

0x8

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

ITS-count**Type**

int

Default value

0x1

Number of Interrupt Translation Services to be instantiated (0=none).

ITS-device-bits**Type**

int

Default value

0x10

Number of bits supported for ITS device IDs.

ITS-enable-itt-address-verification**Type**

bool

Default value

0x0

If true, a transaction will be sent to ITT Address for verification.

ITS-hardware-collection-count**Type**

int

Default value

0x0

Number of hardware collections held exclusively in the ITS.

ITS-shared-vPE-table

Type

int

Default value

0x0

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

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

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

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"

A dotted quad indicating the bitwidth of fields at each affinity level.

allow-LPIEN-clear**Type**

bool

Default value

0x1

Allow RW behaviour on `GICR_CTLR.LPIEN` instead of set once.

chip-count**Type**

int

Default value

0x10

The total number of chips supported.

chip-id**Type**

int

Default value

0x0

Chip ID when multichip operation is enabled.

chip-select-affinity-level**Type**

int

Default value

0x3

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

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

common-vPE-table-affinity**Type**

string

Default value

""

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

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

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

If true, multiple view feature will provide multiple programming views which can be used by multiple hypervisors.

enabled

Type

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

extended-ppi-count**Type**

int

Default value

0x40

Number of extended PPI supported.

fmu-blktype-num**Type**

int

Default value

0x6

Number of stakeholder block types for FMU.

has-gicv4.1**Type**

bool

Default value

0x1

Enable GICv4.1 functionality; when false the component is inactive.

has-two-security-states**Type**

bool

Default value

0x1

If true, has two security states.

max-cores-supported-by-GCI**Type**

int

Default value

0x8

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

max-pe-on-chip**Type**

int

Default value

0x4

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

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"

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

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

GIC-700 dedistributor power management is done by updating GICR_PWRR register.

reg-base**Type**

int

Default value

0x2c010000

GIC-700 base address.

3.10.40 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 3-1050: 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 23.

Changes in 11.27.17

Parameters removed:

- direct-lpi-support
- extended-spi-count

Iris and MTI instances for GIC720AE_Filter

This model has the following Iris instances:

Table 3-1051: 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	ExportTest.GIC720AE_Filterfmu
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_t1	GICv3Distributor

This model has the following MTI trace components:

Table 3-1052: GIC720AE_Filter MTI instances

InstanceName	ComponentName
GIC720AE_Filter	GICv3IRI
GIC720AE_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC720AE_Filter.ITS0	GICv4InterruptTranslationService
GIC720AE_Filter.ITS0.bus_subordinate	PVBusSlave
GIC720AE_Filter.fmu	ExportTest.GIC720AE_Filterfmu
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

GIC720AE_Filter contains the following CADI targets:

- GIC720AE

About GIC720AE_Filter

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_Filter](#).

Ports for GIC720AE_Filter

Table 3-1053: 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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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.
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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
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.

Name	Protocol	Type	Description
fmu_cri	Signal	Master	Critical Interrupt
fmu_eri	Signal	Master	Error recovery Interrupt
icdrdt_out	PVBus	Master	AXI Stream Interface. This is used only when local cross chip addressing is enabled at the moment
icrdtd_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
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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
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

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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 GIC720AE_Filter

CPU-affinities

Type

string

Default value

0.0.0.0

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

""

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

GICD_CTLR-DS-1-means-secure-only

Type

bool

Default value

0x0

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

GICD_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

IIDR**Type**

int

Default value

0x0

GICD_IIDR and GICR_IIDR value.

ITS-ID-bits**Type**

int

Default value

0x10

Number of interrupt bits supported by ITS.

ITS-collection-ID-bits**Type**

int

Default value

0x8

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

ITS-count**Type**

int

Default value

0x1

Number of Interrupt Translation Services to be instantiated (0=none).

ITS-device-bits**Type**

int

Default value

0x10

Number of bits supported for ITS device IDs.

ITS-enable-itt-address-verification**Type**

bool

Default value

0x0

If true, a transaction will be sent to ITT Address for verification.

ITS-hardware-collection-count**Type**

int

Default value

0x0

Number of hardware collections held exclusively in the ITS.

ITS-shared-vPE-table**Type**

int

Default value

0x0

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

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

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

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"

A dotted quad indicating the bitwidth of fields at each affinity level.

allow-LPIEN-clear**Type**

bool

Default value

0x1

Allow RW behaviour on GICR_CTLR.LPIEN instead of set once.

chip-count**Type**

int

Default value

0x10

The total number of chips supported.

chip-id**Type**

int

Default value

0x0

Chip ID when multichip operation is enabled.

chip-select-affinity-level**Type**

int

Default value

0x3

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

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

common-vPE-table-affinity**Type**

string

Default value

""

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

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

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

If true, multiple view feature will provide multiple programming views which can be used by multiple hypervisors.

enabled

Type

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

extended-ppi-count

Type

int

Default value

0x40

Number of extended PPI supported.

fmu-blktype-num

Type

int

Default value

0x6

Number of stakeholder block types for FMU.

has-gicv4.1

Type

bool

Default value

0x1

Enable GICv4.1 functionality; when false the component is inactive.

has-two-security-states**Type**

bool

Default value

0x1

If true, has two security states.

max-cores-supported-by-GCI**Type**

int

Default value

0x8

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

max-pe-on-chip**Type**

int

Default value

0x4

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

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"

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

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

GIC-700 dedistributor power management is done by updating GICR_PWRR register.

reg-base

Type
int

Default value
0x2c010000

GIC-700 base address.

3.10.41 GIC_400

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1054: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for GIC_400

This model has the following Iris instances:

Table 3-1055: GIC_400 Iris instances

InstanceName	ComponentName
GIC_400	GIC_400
GIC_400.vgic_bus_slave	PVBusSlave

This model has the following MTI trace components:

Table 3-1056: GIC_400 MTI instances

InstanceName	ComponentName
GIC_400	GIC_400
GIC_400.vgic_bus_slave	PVBusSlave

GIC_400 contains the following CADI targets:

- GIC_400

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_ISACTIVER_n, GICD_ICACTIVER_n, GICD_ISPENDR_n, or GICD_ICPENDR_n might not work as expected unless there is a configured target in GICD_ICFGR_m.
- 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.

Ports for GIC_400

Table 3-1057: 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

Number of interfaces to support.

NUM_SPIS

Type

int

Default value

0xe0

Number of interrupt pins.

enable_log_errors**Type**

bool

Default value

0x0

.

enable_log_fatal**Type**

bool

Default value

0x0

.

enable_log_warnings**Type**

bool

Default value

0x0

.

interface0.cluster_id**Type**

int

Default value

0x0

The CLUSTERID of the interface you want to appear as interface0 in the VGIC.

interface0.core_id**Type**

int

Default value

0x0

The core id of interface0 in the cluster.

interface0.inout_port_number_to_use**Type**

int

Default value

0x0

Which ppiN port is used for this interface.

interface1.cluster_id**Type**

int

Default value

0x0

The CLUSTERID of the interface you want to appear as interface1 in the VGIC.

interface1.core_id**Type**

int

Default value

0x0

The core id of interface1 in the cluster.

interface1.inout_port_number_to_use**Type**

int

Default value

0x1

Which ppiN port is used for this interface.

interface2.cluster_id**Type**

int

Default value

0x0

The CLUSTERID of the interface you want to appear as interface2 in the VGIC.

interface2.core_id**Type**

int

Default value

0x0

The core id of interface2 in the cluster.

interface2.inout_port_number_to_use**Type**

int

Default value

0x2

Which ppiN port is used for this interface.

interface3.cluster_id**Type**

int

Default value

0x0

The CLUSTERID of the interface you want to appear as interface3 in the VGIC.

interface3.core_id**Type**

int

Default value

0x0

The core id of interface3 in the cluster.

interface3.inout_port_number_to_use**Type**

int

Default value

0x3

Which ppiN port is used for this interface.

interface4.cluster_id**Type**

int

Default value

0x0

The CLUSTERID of the interface you want to appear as interface4 in the VGIC.

interface4.core_id**Type**

int

Default value

0x0

The core id of interface4 in the cluster.

interface4.inout_port_number_to_use**Type**

int

Default value

0x4

Which ppiN port is used for this interface.

interface5.cluster_id**Type**

int

Default value

0x0

The CLUSTERID of the interface you want to appear as interface5 in the VGIC.

interface5.core_id**Type**

int

Default value

0x0

The core id of interface5 in the cluster.

interface5.inout_port_number_to_use**Type**

int

Default value

0x5

Which ppiN port is used for this interface.

interface6.cluster_id**Type**

int

Default value

0x0

The CLUSTERID of the interface you want to appear as interface6 in the VGIC.

interface6.core_id**Type**

int

Default value

0x0

The core id of interface6 in the cluster.

interface6.inout_port_number_to_use**Type**

int

Default value

0x6

Which ppiN port is used for this interface.

interface7.cluster_id**Type**

int

Default value

0x0

The CLUSTERID of the interface you want to appear as interface7 in the VGIC.

interface7.core_id

Type
int

Default value
0x0

The core id of interface7 in the cluster.

interface7.inout_port_number_to_use

Type
int

Default value
0x7

Which ppiN port is used for this interface.

3.10.42 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 3-1058: 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 23.

Iris and MTI instances for GIC_IRI

This model has the following Iris instances:

Table 3-1059: 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 3-1060: GIC_IRI MTI instances

InstanceName	ComponentName
GIC_IRI	GICv3IRI
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

GIC_IRI contains the following CADI targets:

- GIC_IRI

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, `pvbuss_s`, granting access to the register banks used by the configuration and operation of MSIs and the master interface, `pvbuss_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.

Ports for GIC_IRI

Table 3-1061: Ports

Name	Protocol	Type	Description
<code>cfgsdisable</code>	Signal	Slave	Disable some SPIs signal.
<code>extended_ppi_in_0[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.
<code>extended_ppi_in_1[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.
<code>extended_ppi_in_10[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 10.
<code>extended_ppi_in_100[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100.
<code>extended_ppi_in_101[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101.
<code>extended_ppi_in_102[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102.
<code>extended_ppi_in_103[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103.
<code>extended_ppi_in_104[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104.
<code>extended_ppi_in_105[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105.
<code>extended_ppi_in_106[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106.
<code>extended_ppi_in_107[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107.
<code>extended_ppi_in_108[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108.
<code>extended_ppi_in_109[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 109.
<code>extended_ppi_in_11[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 11.
<code>extended_ppi_in_110[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 110.
<code>extended_ppi_in_111[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 111.
<code>extended_ppi_in_112[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 112.

Name	Protocol	Type	Description
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.
extended_ppi_in_133[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 133.

Name	Protocol	Type	Description
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.
extended_ppi_in_154[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 154.

Name	Protocol	Type	Description
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.
extended_ppi_in_175[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 175.

Name	Protocol	Type	Description
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.
extended_ppi_in_196[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 196.

Name	Protocol	Type	Description
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.
extended_ppi_in_216[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 216.

Name	Protocol	Type	Description
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.
extended_ppi_in_237[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 237.

Name	Protocol	Type	Description
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.
extended_ppi_in_28[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 28.

Name	Protocol	Type	Description
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.
extended_ppi_in_49[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 49.

Name	Protocol	Type	Description
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.
extended_ppi_in_7[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 7.

Name	Protocol	Type	Description
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.
extended_ppi_in_90[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 90.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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[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

Device supports affinity level 3 values that are non-zero.

ARE-fixed-to-one

Type

bool

Default value

0x0

GICv2 compatibility is not supported and GICD_CTLR.ARE_* is always one.

CPU-affinities

Type

string

Default value

0.0.0.0

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

""

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

Limit application of DPG bits to interrupt groups for which ARE=1.

DPG-bits-implemented**Type**

bool

Default value

0x0

Enable implementation of interrupt group participation bits or DPG bits in GICR_CTLR.

DS-fixed-to-zero**Type**

bool

Default value

0x0

Enable/disable support of single security state.

GICD-alias**Type**

int

Default value

0x0

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

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

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

If true, the GICD_ITARGETS registers are RAZ/WI.

GICD_PIDR

Type

int

Default value

0x0

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

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

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

When true, the registers GICR_INVLPIR, GICR_INVALLR and GICR_SYNCRR are implemented.

GICR_PIDR**Type**

int

Default value

0x0

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

GICR_PROPBASER register is read-only.

GICR_PROPBASER-reset-value**Type**

int

Default value

0x0

Value of GICR_PROPBASER on reset.

GITS_BASER0-entry-bytes**Type**

int

Default value

0x8

Number of bytes required per entry for GITS_BASER0 register.

GITS_BASER0-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER0 register is RAZ/WI.

GITS_BASER0-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER1 register.

GITS_BASER1-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER1 register is RAZ/WI.

GITS_BASER1-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER2 register.

GITS_BASER2-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER2 register is RAZ/WI.

GITS_BASER2-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER3 register.

GITS_BASER3-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER3 register is RAZ/WI.

GITS_BASER3-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER4 register.

GITS_BASER4-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER4 register is RAZ/WI.

GITS_BASER4-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER5 register.

GITS_BASER5-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER5 register is RAZ/WI.

GITS_BASER5-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER6 register.

GITS_BASER6-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER6 register is RAZ/WI.

GITS_BASER6-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER7 register.

GITS_BASER7-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER7 register is RAZ/WI.

GITS_BASER7-type**Type**

int

Default value

0x0

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

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

Mask for writes to ICFGR registers that configure PPIs.

ICFGR-PPI-reset**Type**

int

Default value

0x0

Reset value for ICFGR registers that configure PPIs.

ICFGR-SGI-mask**Type**

int

Default value

0x0

Mask for writes to ICFGR registers that configure SGIs.

ICFGR-SGI-reset**Type**

int

Default value

0xaaaaaaaa

Reset value for ICFGR registers that configure SGIs.

ICFGR-SPI-mask**Type**

int

Default value

0xaaaaaaaa

Mask for writes to ICFGR registers that configure SPIs.

ICFGR-SPI-reset**Type**

int

Default value

0x0

Reset value for ICFGR registers that configure SPIs.

ICFGR-rsvd-bit**Type**

bool

Default value

0x0

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGR_n for n>0.

IGROUP-PPI-mask**Type**

int

Default value

0xffff

Mask for writes to PPI bits in IGROUP registers.

IGROUP-PPI-reset**Type**

int

Default value

0x0

Reset value for SGI bits in IGROUP registers.

IGROUP-SGI-mask**Type**

int

Default value

0xffff

Mask for writes to SGI bits in IGROUP registers.

IGROUP-SGI-reset**Type**

int

Default value

0x0

Reset value for SGI bits in IGROUP registers.

IIDR**Type**

int

Default value

0x0

GICD_IIDR and GICR_IIDR value.

IRI-ID-bits**Type**

int

Default value

0x10

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

GICD_IROUTERn.InterruptRoutingMode is RAZ/WI.

ITS-BASER-force-page-alignment**Type**

bool

Default value

0x1

Force alignment of address written to a GITS_BASER register to the page size configured.

ITS-ID-bits**Type**

int

Default value

0x10

Number of interrupt bits supported by ITS.

ITS-MOVALL-update-collections**Type**

bool

Default value

0x0

Whether MOVALL command updates the collection entires.

ITS-TRANSLATE64R**Type**

bool

Default value

0x0

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

Sets the RO bit GITS_TYPER.INV. When true, after the following sequence: 1) GITS_CTLR.Enabled written 1-->0, 2) GITS_CTLR.Quirescent 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

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0.

ITS-count**Type**

int

Default value

0x0

Number of Interrupt Translation Services to be instantiated (0=none).

ITS-cumulative-collection-tables**Type**

bool

Default value

0x1

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

Number of bits supported for ITS device IDs.

ITS-enable-itt-address-verification**Type**

bool

Default value

0x0

If true, a transaction will be sent to ITT Address for verification.

ITS-entry-size**Type**

int

Default value

0x8

Number of bytes required to store each entry in the ITT tables.

ITS-hardware-collection-count**Type**

int

Default value

0x0

Number of hardware collections held exclusively in the ITS.

ITS-legacy-iidr-typer-offset**Type**

bool

Default value

0x0

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

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

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

Use physical hardware addresses for targets in ITS commands -- must be true for distributed implementations.

ITS-vmovp-bit**Type**

bool

Default value

0x0

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

Register base address for ITS0 (automatic if 0).

ITS1-base**Type**

int

Default value

0x0

Register base address for ITS1 (automatic if 0).

ITS2-base**Type**

int

Default value

0x0

Register base address for ITS2 (automatic if 0).

ITS3-base**Type**

int

Default value

0x0

Register base address for ITS3 (automatic if 0).

LPI-cache-check-data**Type**

bool

Default value

0x0

Enable Cached LPI data against memory checking when available for cache type.

LPI-cache-type**Type**

int

Default value

0x1

Cache type for LPIs, 0:No caching, 1:Full caching.

MSI_IIDR**Type**

int

Default value

0x0

Value returned in MSI_IIDR registers.

MSI_NS-frame0-base**Type**

int

Default value

0x0

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Minimum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

MSI_PIDR**Type**

int

Default value

0x0

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

If non-zero, sets the base address used for secure MSI frame 0 registers.

MSI_S-frame0-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

MSI_S-frame1-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 1 registers.

MSI_S-frame1-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

MSI_S-frame2-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 2 registers.

MSI_S-frame2-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

MSI_S-frame3-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 3 registers.

MSI_S-frame3-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

MSI_S-frame4-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 4 registers.

MSI_S-frame4-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

MSI_S-frame5-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 5 registers.

MSI_S-frame5-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

MSI_S-frame6-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 6 registers.

MSI_S-frame6-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

MSI_S-frame7-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 7 registers.

MSI_S-frame7-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

PA_SIZE**Type**

int

Default value

0x30

Number of valid bits in physical address.

PPI-implemented-mask**Type**

int

Default value

0xffff

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

Number of SPIs that are implemented.

SPI-message-based-support**Type**

bool

Default value

0x1

Distributor supports message based signaling of SPI.

SPI-unimplemented**Type**

string

Default value

""

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

Determines whether the GICR_STATUSR register is implemented.

add-output-cpu-wake-request-signal-from-redistributor**Type**

bool

Default value

0x0

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

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

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

common-lpi-configuration**Type**

int

Default value

0x0

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

""

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, [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

Delay accesses from the ITS until GICR_SYNCNR is read.

delay-redistributor-accesses**Type**

bool

Default value

0x1

Delay memory accesses from the redistributor until GICR_SYNCNR is read.

direct-lpi-support**Type**

bool

Default value

0x0

Enable support for LPI operations through GICR registers.

enable_protocol_checking**Type**

bool

Default value

0x0

Enable/disable protocol checking at cpu interface.

enabled**Type**

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

extended-ppi-count**Type**

int

Default value

0x0

Number of extended PPI supported.

extended-spi-count

Type

int

Default value

0x0

Number of extended SPI supported.

fixed-routed-spis

Type

string

Default value

""

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

Number of extended GICR_ICFGR registers supported.

gicv2-only

Type

bool

Default value

0x0

If true, when using the GICv3 model, pretend to be a GICv2 system.

group-enables-control-doorbell

Type

bool

Default value

0x0

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

Enable GICv4.1 functionality; when false the component is inactive.

has-two-security-states**Type**

bool

Default value

0x1

If true, has two security states.

has_VPENDBASER-dirty-flag-on-load**Type**

bool

Default value

0x0

GICR_VPENDBASER.Dirty reflects transient loading state when valid=1.

has_mpam**Type**

bool

Default value

0x0

Enable MPAM support on ITS and RDs.

has_nmi**Type**

bool

Default value

0x0

Enable support for Non-maskable Interrupts (NMIs). (FEAT_GICv3_NMI).

ignore-generate-sgi-when-no-are**Type**

bool

Default value

0x0

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

For IRI with support of virtual interrupt, individual doorbell is not supported when true.

irouter-default-mask**Type**

string

Default value

""

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

irouter-default-reset**Type**

string

Default value

""

Default Reset Value of IROUTER[32..1019] register in the form 'a.b.c.d' or '*'.

irouter-mask-values**Type**

string

Default value

""

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

""

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

Enables for SGI associated with an ARE=0 regime are RAO/WI.

local-SEIs**Type**

bool

Default value

0x0

Generate SEI to signal internal issues.

local-VSEIs**Type**

bool

Default value

0x0

Generate VSEI to signal internal issues.

lockable-SPI-count**Type**

int

Default value

0x0

Number of SPIs that are locked down when CFGSDISABLE signal is asserted. Only applies for GICv2.

monolithic**Type**

bool

Default value

0x0

Indicate that the implementation is not distributed.

mpam_max_partid**Type**

int

Default value

0xffff

Maximum valid PARTID.

mpam_max_pmg**Type**

int

Default value

0xff

Maximum valid PMG.

non-ARE-core-count**Type**

int

Default value

0x8

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

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"
```

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

Print memory map to stdout.

priority-bits**Type**

int

Default value

0x5

Number of implemented priority bits.

processor-numbers**Type**

string

Default value

""

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

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

reg-base**Type**

int

Default value

0x2c010000

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

""

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

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

Default MECID value for NON-SECURE PAS.

sgi-range-selector-support**Type**

bool

Default value

0x0

Device has support for the Range Selector feature for SGI.

single-set-support**Type**

bool

Default value

0x0

When true, forces redistributors to recall interrupts with a clear rather than issue a second Set command.

supports-shareability**Type**

bool

Default value

0x1

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

Trace LPI property updates performed on speculative accesses (useful for debugging LPI).

vPE-table-entry-size-in-doubleword**Type**

int

Default value

0x5

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

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.

virtual-priority-bits**Type**

int

Default value

0x5

Number of implemented virtual priority bits.

wakeup-on-reset**Type**

bool

Default value

0x0

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.

3.10.43 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 3-1062: 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 23.

Iris and MTI instances for GIC_IRI_Filter

This model has the following Iris instances:

Table 3-1063: 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_tl	GICv3Distributor

This model has the following MTI trace components:

Table 3-1064: GIC_IRI_Filter MTI instances

InstanceName	ComponentName
GIC_IRI_Filter	GICv3IRI
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

GIC_IRI_Filter contains the following CADI targets:

- GIC_IRI

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 `pvcus_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.

Ports for GIC_IRI_Filter

Table 3-1065: 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.
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.

Name	Protocol	Type	Description
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.
extended_ppi_in_137[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 137.

Name	Protocol	Type	Description
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.
extended_ppi_in_158[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 158.

Name	Protocol	Type	Description
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.
extended_ppi_in_179[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 179.

Name	Protocol	Type	Description
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.
extended_ppi_in_2[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 2.

Name	Protocol	Type	Description
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.
extended_ppi_in_22[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 22.

Name	Protocol	Type	Description
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.
extended_ppi_in_240[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 240.

Name	Protocol	Type	Description
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.
extended_ppi_in_31[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 31.

Name	Protocol	Type	Description
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.
extended_ppi_in_52[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 52.

Name	Protocol	Type	Description
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.
extended_ppi_in_73[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 73.

Name	Protocol	Type	Description
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.
extended_ppi_in_94[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 94.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
pvbuss_filtermiss_m	PVBus	Master	Passthrough for accesses to pages not used by the GIC IRI.
pvbuss_m	PVBus	Master	Memory bus for transactions generated by the GIC.
pvbuss_s	PVBus	Slave	Memory bus in.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.

Name	Protocol	Type	Description
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

Device supports affinity level 3 values that are non-zero.

ARE-fixed-to-one

Type

bool

Default value

0x0

GICv2 compatibility is not supported and GICD_CTLR.ARE_* is always one.

CPU-affinities

Type

string

Default value

0.0.0.0

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

""

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

Limit application of DPG bits to interrupt groups for which ARE=1.

DPG-bits-implemented**Type**

bool

Default value

0x0

Enable implementation of interrupt group participation bits or DPG bits in GICR_CTLR.

DS-fixed-to-zero**Type**

bool

Default value

0x0

Enable/disable support of single security state.

GICD-alias**Type**

int

Default value

0x0

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

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

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

If true, the GICD_ITARGETS registers are RAZ/WI.

GICD_PIDR**Type**

int

Default value

0x0

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

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

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

When true, the registers GICR_INVLPIR, GICR_INVALLR and GICR_SYNCR are implemented.

GICR_PIDR**Type**

int

Default value

0x0

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

GICR_PROPBASER register is read-only.

GICR_PROPBASER-reset-value**Type**

int

Default value

0x0

Value of GICR_PROPBASER on reset.

GITS_BASER0-entry-bytes**Type**

int

Default value

0x8

Number of bytes required per entry for GITS_BASER0 register.

GITS_BASER0-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER0 register is RAZ/WI.

GITS_BASER0-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER1 register.

GITS_BASER1-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER1 register is RAZ/WI.

GITS_BASER1-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER2 register.

GITS_BASER2-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER2 register is RAZ/WI.

GITS_BASER2-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER3 register.

GITS_BASER3-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER3 register is RAZ/WI.

GITS_BASER3-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER4 register.

GITS_BASER4-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER4 register is RAZ/WI.

GITS_BASER4-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER5 register.

GITS_BASER5-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER5 register is RAZ/WI.

GITS_BASER5-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER6 register.

GITS_BASER6-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER6 register is RAZ/WI.

GITS_BASER6-type**Type**

int

Default value

0x0

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

Number of bytes required per entry for GITS_BASER7 register.

GITS_BASER7-indirect-RAZ**Type**

bool

Default value

0x0

Indirect field for GITS_BASER7 register is RAZ/WI.

GITS_BASER7-type**Type**

int

Default value

0x0

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

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

Mask for writes to ICFGR registers that configure PPIs.

ICFGR-PPI-reset**Type**

int

Default value

0x0

Reset value for ICFGR registers that configure PPIs.

ICFGR-SGI-mask**Type**

int

Default value

0x0

Mask for writes to ICFGR registers that configure SGIs.

ICFGR-SGI-reset**Type**

int

Default value

0xaaaaaaaa

Reset value for ICFGR registers that configure SGIs.

ICFGR-SPI-mask**Type**

int

Default value

0xaaaaaaaa

Mask for writes to ICFGR registers that configure SPIs.

ICFGR-SPI-reset**Type**

int

Default value

0x0

Reset value for ICFGR registers that configure SPIs.

ICFGR-rsvd-bit**Type**

bool

Default value

0x0

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGR_n for n>0.

IGROUP-PPI-mask**Type**

int

Default value

0xffff

Mask for writes to PPI bits in IGROUP registers.

IGROUP-PPI-reset**Type**

int

Default value

0x0

Reset value for SGI bits in IGROUP registers.

IGROUP-SGI-mask**Type**

int

Default value

0xffff

Mask for writes to SGI bits in IGROUP registers.

IGROUP-SGI-reset**Type**

int

Default value

0x0

Reset value for SGI bits in IGROUP registers.

IIDR**Type**

int

Default value

0x0

GICD_IIDR and GICR_IIDR value.

IRI-ID-bits**Type**

int

Default value

0x10

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

GICD_IROUTERn.InterruptRoutingMode is RAZ/WI.

ITS-BASER-force-page-alignment**Type**

bool

Default value

0x1

Force alignment of address written to a GITS_BASER register to the page size configured.

ITS-ID-bits**Type**

int

Default value

0x10

Number of interrupt bits supported by ITS.

ITS-MOVALL-update-collections**Type**

bool

Default value

0x0

Whether MOVALL command updates the collection entires.

ITS-TRANSLATE64R**Type**

bool

Default value

0x0

Add an implementation specific register at 0x10008 supporting 64 bit TRANSLATER (dev[63:32], interupt[31:0]).

ITS-cache-invalidate-on-disable**Type**

bool

Default value

0x0

Sets the RO bit GITS_TYPER.INV. When true, after the following sequence: 1) GITS_CTLR.Enabled written 1-->0, 2) GITS_CTLR.Quirescent 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

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0.

ITS-count**Type**

int

Default value

0x0

Number of Interrupt Translation Services to be instantiated (0=none).

ITS-cumulative-collection-tables**Type**

bool

Default value

0x1

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

Number of bits supported for ITS device IDs.

ITS-enable-itt-address-verification**Type**

bool

Default value

0x0

If true, a transaction will be sent to ITT Address for verification.

ITS-entry-size**Type**

int

Default value

0x8

Number of bytes required to store each entry in the ITT tables.

ITS-hardware-collection-count**Type**

int

Default value

0x0

Number of hardware collections held exclusively in the ITS.

ITS-legacy-iidr-typer-offset**Type**

bool

Default value

0x0

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

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

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

Use physical hardware addresses for targets in ITS commands -- must be true for distributed implementations.

ITS-vmovp-bit**Type**

bool

Default value

0x0

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

Register base address for ITS0 (automatic if 0).

ITS1-base**Type**

int

Default value

0x0

Register base address for ITS1 (automatic if 0).

ITS2-base**Type**

int

Default value

0x0

Register base address for ITS2 (automatic if 0).

ITS3-base**Type**

int

Default value

0x0

Register base address for ITS3 (automatic if 0).

LPI-cache-check-data**Type**

bool

Default value

0x0

Enable Cached LPI data against memory checking when available for cache type.

LPI-cache-type**Type**

int

Default value

0x1

Cache type for LPIs, 0:No caching, 1:Full caching.

MSI_IIDR**Type**

int

Default value

0x0

Value returned in MSI_IIDR registers.

MSI_NS-frame0-base**Type**

int

Default value

0x0

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Minimum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

MSI_PIDR**Type**

int

Default value

0x0

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

If non-zero, sets the base address used for secure MSI frame 0 registers.

MSI_S-frame0-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

MSI_S-frame1-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 1 registers.

MSI_S-frame1-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

MSI_S-frame2-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 2 registers.

MSI_S-frame2-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

MSI_S-frame3-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 3 registers.

MSI_S-frame3-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

MSI_S-frame4-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 4 registers.

MSI_S-frame4-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

MSI_S-frame5-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 5 registers.

MSI_S-frame5-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

MSI_S-frame6-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 6 registers.

MSI_S-frame6-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

MSI_S-frame7-base**Type**

int

Default value

0x0

If non-zero, sets the base address used for secure MSI frame 7 registers.

MSI_S-frame7-max-SPI**Type**

int

Default value

0x0

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

Minimum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

PA_SIZE**Type**

int

Default value

0x30

Number of valid bits in physical address.

PPI-implemented-mask**Type**

int

Default value

0xffff

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

Number of SPIs that are implemented.

SPI-message-based-support**Type**

bool

Default value

0x1

Distributor supports message based signaling of SPI.

SPI-unimplemented**Type**

string

Default value

""

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

Determines whether the GICR_STATUSR register is implemented.

add-output-cpu-wake-request-signal-from-redistributor**Type**

bool

Default value

0x0

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

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

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

common-lpi-configuration**Type**

int

Default value

0x0

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

""

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, [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

Delay accesses from the ITS until GICR_SYNCRR is read.

delay-redistributor-accesses**Type**

bool

Default value

0x1

Delay memory accesses from the redistributor until GICR_SYNCRR is read.

direct-lpi-support**Type**

bool

Default value

0x0

Enable support for LPI operations through GICR registers.

enable_protocol_checking**Type**

bool

Default value

0x0

Enable/disable protocol checking at cpu interface.

enabled**Type**

bool

Default value

0x1

Enable GICv3 functionality; when false the component is inactive.

extended-ppi-count**Type**

int

Default value

0x0

Number of extended PPI supported.

extended-spi-count

Type

int

Default value

0x0

Number of extended SPI supported.

fixed-routed-spis

Type

string

Default value

""

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

Number of extended GICR_ICFGR registers supported.

gicv2-only

Type

bool

Default value

0x0

If true, when using the GICv3 model, pretend to be a GICv2 system.

group-enables-control-doorbell

Type

bool

Default value

0x0

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

Enable GICv4.1 functionality; when false the component is inactive.

has-two-security-states**Type**

bool

Default value

0x1

If true, has two security states.

has_VPENDBASER-dirty-flag-on-load**Type**

bool

Default value

0x0

GICR_VPENDBASER.Dirty reflects transient loading state when valid=1.

has_mpam**Type**

bool

Default value

0x0

Enable MPAM support on ITS and RDs.

has_nmi**Type**

bool

Default value

0x0

Enable support for Non-maskable Interrupts (NMIs). (FEAT_GICv3_NMI).

ignore-generate-sgi-when-no-are**Type**

bool

Default value

0x0

Ignore GenerateSGI packets coming form the CPU interface if both ARE_S and ARE_NS are 0.

individual-doorbell-not-supported**Type**

bool

Default value

0x0

For IRI with support of virtual interrupt, individual doorbell is not supported when true.

irouter-default-mask**Type**

string

Default value

""

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

irouter-default-reset**Type**

string

Default value

""

Default Reset Value of IROUTER[32..1019] register in the form 'a.b.c.d' or *.

irouter-mask-values**Type**

string

Default value

""

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

""

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

Enables for SGI associated with an ARE=0 regime are RAO/WI.

local-SEIs**Type**

bool

Default value

0x0

Generate SEI to signal internal issues.

local-VSEIs**Type**

bool

Default value

0x0

Generate VSEI to signal internal issues.

lockable-SPI-count**Type**

int

Default value

0x0

Number of SPIs that are locked down when CFGSDISABLE signal is asserted. Only applies for GICv2.

monolithic**Type**

bool

Default value

0x0

Indicate that the implementation is not distributed.

mpam_max_partid**Type**

int

Default value

0xffff

Maximum valid PARTID.

mpam_max_pmg**Type**

int

Default value

0xff

Maximum valid PMG.

non-ARE-core-count**Type**

int

Default value

0x8

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

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"
```

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

Print memory map to stdout.

priority-bits**Type**

int

Default value

0x5

Number of implemented priority bits.

processor-numbers**Type**

string

Default value

""

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

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

reg-base**Type**

int

Default value

0x2c010000

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

""

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

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

Default MECID value for NON-SECURE PAS.

sgi-range-selector-support**Type**

bool

Default value

0x0

Device has support for the Range Selector feature for SGI.

single-set-support**Type**

bool

Default value

0x0

When true, forces redistributors to recall interrupts with a clear rather than issue a second Set command.

supports-shareability**Type**

bool

Default value

0x1

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

Trace LPI property updates performed on speculative accesses (useful for debugging LPI).

vPE-table-entry-size-in-doubleword**Type**

int

Default value

0x5

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

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.

virtual-priority-bits**Type**

int

Default value

0x5

Number of implemented virtual priority bits.

wakeup-on-reset**Type**

bool

Default value

0x0

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.

3.10.44 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 3-1066: 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 23.

Iris and MTI instances for ICS307

This model has the following Iris instances:

Table 3-1067: ICS307 Iris instances

InstanceName	ComponentName
ICS307	ICS307
ICS307.clkdiv_clk1	ClockDivider

This model has the following MTI trace components:

Table 3-1068: ICS307 MTI instances

InstanceName	ComponentName
ICS307.clkdiv_clk1	ClockDivider

ICS307 contains the following CADI targets:

- ClockDivider
- ICS307

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:

```
Divisor = ((rdw+2) * scale) / (2 * (vdw+8))
```

where `scale` is derived from this table indexed by `od`:

Table 3-1069: od to scale conversion

od	scale
0	10
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:

$$((6+2) * 4) / (2 * (4+8)) = 4/3$$

Ports for ICS307

Table 3-1070: 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

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

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

od
Type
int
Default value
0x3

OD.

rdr
Type
int
Default value
0x6

RDR.

vdw
Type
int
Default value
0x4

VDW.

3.10.45 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 3-1071: 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 23.

Iris and MTI instances for IDAU

This model has the following Iris instances:

Table 3-1072: IDAU Iris instances

InstanceName	ComponentName
IDAU.bus_bridge	PVBusBridge

IDAU interface

- input `IDAUADDR` Address of the region
- output `IDAUNS` Non-secure region response
- output `IDAUNSC` Non-secure-callable region response
- output `IDAUID` Region number
- output `IDAUIDV` Region number valid
- output `IDAUNCHK` Region exempt from attribution check

The Implementation Defined Attribution Unit (IDAU) model uses the `pv::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 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.

Ports for IDAU

Table 3-1073: 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

Base address of IDAU region0.

IDAU_REGION0.ENABLE

Type

bool

Default value

0x0

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

Mark IDAU region0 as exempt.

IDAU_REGION0.LADDR

Type

int

Default value

0x0

Limit address of IDAU region0.

IDAU_REGION0.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region1.

IDAU_REGION1.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region1 as exempt.

IDAU_REGION1.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region1.

IDAU_REGION1.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region10.

IDAU_REGION10.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region10 as exempt.

IDAU_REGION10.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region10.

IDAU_REGION10.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region100.

IDAU_REGION100.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region100 as exempt.

IDAU_REGION100.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region100.

IDAU_REGION100.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region101.

IDAU_REGION101.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region101 as exempt.

IDAU_REGION101.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region101.

IDAU_REGION101.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region102.

IDAU_REGION102.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region102 as exempt.

IDAU_REGION102.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region102.

IDAU_REGION102.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region103.

IDAU_REGION103.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region103 as exempt.

IDAU_REGION103.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region103.

IDAU_REGION103.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region104.

IDAU_REGION104.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region104 as exempt.

IDAU_REGION104.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region104.

IDAU_REGION104.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region105.

IDAU_REGION105.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region105 as exempt.

IDAU_REGION105.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region105.

IDAU_REGION105.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region106.

IDAU_REGION106.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region106 as exempt.

IDAU_REGION106.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region106.

IDAU_REGION106.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region107.

IDAU_REGION107.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region107 as exempt.

IDAU_REGION107.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region107.

IDAU_REGION107.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region108.

IDAU_REGION108.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region108 as exempt.

IDAU_REGION108.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region108.

IDAU_REGION108.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region109.

IDAU_REGION109.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region109 as exempt.

IDAU_REGION109.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region109.

IDAU_REGION109.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region11.

IDAU_REGION11.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region11 as exempt.

IDAU_REGION11.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region11.

IDAU_REGION11.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region110.

IDAU_REGION110.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region110 as exempt.

IDAU_REGION110.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region110.

IDAU_REGION110.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region111.

IDAU_REGION111.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region111 as exempt.

IDAU_REGION111.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region111.

IDAU_REGION111.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region112.

IDAU_REGION112.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region112 as exempt.

IDAU_REGION112.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region112.

IDAU_REGION112.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region113.

IDAU_REGION113.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region113 as exempt.

IDAU_REGION113.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region113.

IDAU_REGION113.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region114.

IDAU_REGION114.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region114 as exempt.

IDAU_REGION114.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region114.

IDAU_REGION114.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region115.

IDAU_REGION115.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region115 as exempt.

IDAU_REGION115.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region115.

IDAU_REGION115.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region116.

IDAU_REGION116.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region116 as exempt.

IDAU_REGION116.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region116.

IDAU_REGION116.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region117.

IDAU_REGION117.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region117 as exempt.

IDAU_REGION117.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region117.

IDAU_REGION117.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region118.

IDAU_REGION118.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region118 as exempt.

IDAU_REGION118.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region118.

IDAU_REGION118.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region119.

IDAU_REGION119.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region119 as exempt.

IDAU_REGION119.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region119.

IDAU_REGION119.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region12.

IDAU_REGION12.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region12 as exempt.

IDAU_REGION12.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region12.

IDAU_REGION12.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region120.

IDAU_REGION120.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region120 as exempt.

IDAU_REGION120.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region120.

IDAU_REGION120.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region121.

IDAU_REGION121.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region121 as exempt.

IDAU_REGION121.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region121.

IDAU_REGION121.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region122.

IDAU_REGION122.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region122 as exempt.

IDAU_REGION122.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region122.

IDAU_REGION122.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region123.

IDAU_REGION123.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region123 as exempt.

IDAU_REGION123.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region123.

IDAU_REGION123.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region124.

IDAU_REGION124.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region124 as exempt.

IDAU_REGION124.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region124.

IDAU_REGION124.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region125.

IDAU_REGION125.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region125 as exempt.

IDAU_REGION125.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region125.

IDAU_REGION125.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region126.

IDAU_REGION126.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region126 as exempt.

IDAU_REGION126.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region126.

IDAU_REGION126.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region127.

IDAU_REGION127.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region127 as exempt.

IDAU_REGION127.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region127.

IDAU_REGION127.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region128.

IDAU_REGION128.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region128 as exempt.

IDAU_REGION128.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region128.

IDAU_REGION128.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region129.

IDAU_REGION129.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region129 as exempt.

IDAU_REGION129.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region129.

IDAU_REGION129.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region13.

IDAU_REGION13.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region13 as exempt.

IDAU_REGION13.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region13.

IDAU_REGION13.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region130.

IDAU_REGION130.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region130 as exempt.

IDAU_REGION130.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region130.

IDAU_REGION130.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region131.

IDAU_REGION131.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region131 as exempt.

IDAU_REGION131.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region131.

IDAU_REGION131.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region132.

IDAU_REGION132.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region132 as exempt.

IDAU_REGION132.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region132.

IDAU_REGION132.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region133.

IDAU_REGION133.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region133 as exempt.

IDAU_REGION133.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region133.

IDAU_REGION133.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region134.

IDAU_REGION134.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region134 as exempt.

IDAU_REGION134.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region134.

IDAU_REGION134.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region135.

IDAU_REGION135.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region135 as exempt.

IDAU_REGION135.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region135.

IDAU_REGION135.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region136.

IDAU_REGION136.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region136 as exempt.

IDAU_REGION136.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region136.

IDAU_REGION136.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region137.

IDAU_REGION137.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region137 as exempt.

IDAU_REGION137.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region137.

IDAU_REGION137.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region138.

IDAU_REGION138.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region138 as exempt.

IDAU_REGION138.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region138.

IDAU_REGION138.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region139.

IDAU_REGION139.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region139 as exempt.

IDAU_REGION139.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region139.

IDAU_REGION139.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region14.

IDAU_REGION14.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region14 as exempt.

IDAU_REGION14.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region14.

IDAU_REGION14.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region140.

IDAU_REGION140.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region140 as exempt.

IDAU_REGION140.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region140.

IDAU_REGION140.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region141.

IDAU_REGION141.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region141 as exempt.

IDAU_REGION141.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region141.

IDAU_REGION141.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region142.

IDAU_REGION142.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region142 as exempt.

IDAU_REGION142.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region142.

IDAU_REGION142.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region143.

IDAU_REGION143.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region143 as exempt.

IDAU_REGION143.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region143.

IDAU_REGION143.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region144.

IDAU_REGION144.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region144 as exempt.

IDAU_REGION144.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region144.

IDAU_REGION144.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region145.

IDAU_REGION145.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region145 as exempt.

IDAU_REGION145.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region145.

IDAU_REGION145.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region146.

IDAU_REGION146.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region146 as exempt.

IDAU_REGION146.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region146.

IDAU_REGION146.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region147.

IDAU_REGION147.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region147 as exempt.

IDAU_REGION147.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region147.

IDAU_REGION147.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region148.

IDAU_REGION148.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region148 as exempt.

IDAU_REGION148.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region148.

IDAU_REGION148.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region149.

IDAU_REGION149.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region149 as exempt.

IDAU_REGION149.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region149.

IDAU_REGION149.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region15.

IDAU_REGION15.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region15 as exempt.

IDAU_REGION15.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region15.

IDAU_REGION15.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region150.

IDAU_REGION150.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region150 as exempt.

IDAU_REGION150.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region150.

IDAU_REGION150.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region151.

IDAU_REGION151.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region151 as exempt.

IDAU_REGION151.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region151.

IDAU_REGION151.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region152.

IDAU_REGION152.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region152 as exempt.

IDAU_REGION152.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region152.

IDAU_REGION152.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region153.

IDAU_REGION153.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region153 as exempt.

IDAU_REGION153.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region153.

IDAU_REGION153.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region154.

IDAU_REGION154.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region154 as exempt.

IDAU_REGION154.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region154.

IDAU_REGION154.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region155.

IDAU_REGION155.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region155 as exempt.

IDAU_REGION155.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region155.

IDAU_REGION155.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region156.

IDAU_REGION156.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region156 as exempt.

IDAU_REGION156.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region156.

IDAU_REGION156.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region157.

IDAU_REGION157.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region157 as exempt.

IDAU_REGION157.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region157.

IDAU_REGION157.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region158.

IDAU_REGION158.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region158 as exempt.

IDAU_REGION158.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region158.

IDAU_REGION158.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region159.

IDAU_REGION159.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region159 as exempt.

IDAU_REGION159.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region159.

IDAU_REGION159.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region16.

IDAU_REGION16.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region16 as exempt.

IDAU_REGION16.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region16.

IDAU_REGION16.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region160.

IDAU_REGION160.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region160 as exempt.

IDAU_REGION160.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region160.

IDAU_REGION160.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region161.

IDAU_REGION161.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region161 as exempt.

IDAU_REGION161.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region161.

IDAU_REGION161.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region162.

IDAU_REGION162.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region162 as exempt.

IDAU_REGION162.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region162.

IDAU_REGION162.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region163.

IDAU_REGION163.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region163 as exempt.

IDAU_REGION163.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region163.

IDAU_REGION163.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region164.

IDAU_REGION164.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region164 as exempt.

IDAU_REGION164.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region164.

IDAU_REGION164.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region165.

IDAU_REGION165.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region165 as exempt.

IDAU_REGION165.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region165.

IDAU_REGION165.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region166.

IDAU_REGION166.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region166 as exempt.

IDAU_REGION166.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region166.

IDAU_REGION166.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region167.

IDAU_REGION167.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region167 as exempt.

IDAU_REGION167.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region167.

IDAU_REGION167.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region168.

IDAU_REGION168.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region168 as exempt.

IDAU_REGION168.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region168.

IDAU_REGION168.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region169.

IDAU_REGION169.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region169 as exempt.

IDAU_REGION169.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region169.

IDAU_REGION169.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region17.

IDAU_REGION17.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region17 as exempt.

IDAU_REGION17.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region17.

IDAU_REGION17.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region170.

IDAU_REGION170.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region170 as exempt.

IDAU_REGION170.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region170.

IDAU_REGION170.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region171.

IDAU_REGION171.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region171 as exempt.

IDAU_REGION171.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region171.

IDAU_REGION171.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region172.

IDAU_REGION172.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region172 as exempt.

IDAU_REGION172.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region172.

IDAU_REGION172.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region173.

IDAU_REGION173.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region173 as exempt.

IDAU_REGION173.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region173.

IDAU_REGION173.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region174.

IDAU_REGION174.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region174 as exempt.

IDAU_REGION174.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region174.

IDAU_REGION174.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region175.

IDAU_REGION175.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region175 as exempt.

IDAU_REGION175.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region175.

IDAU_REGION175.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region176.

IDAU_REGION176.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region176 as exempt.

IDAU_REGION176.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region176.

IDAU_REGION176.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region177.

IDAU_REGION177.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region177 as exempt.

IDAU_REGION177.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region177.

IDAU_REGION177.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region178.

IDAU_REGION178.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region178 as exempt.

IDAU_REGION178.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region178.

IDAU_REGION178.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region179.

IDAU_REGION179.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region179 as exempt.

IDAU_REGION179.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region179.

IDAU_REGION179.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region18.

IDAU_REGION18.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region18 as exempt.

IDAU_REGION18.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region18.

IDAU_REGION18.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region180.

IDAU_REGION180.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region180 as exempt.

IDAU_REGION180.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region180.

IDAU_REGION180.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region181.

IDAU_REGION181.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region181 as exempt.

IDAU_REGION181.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region181.

IDAU_REGION181.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region182.

IDAU_REGION182.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region182 as exempt.

IDAU_REGION182.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region182.

IDAU_REGION182.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region183.

IDAU_REGION183.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region183 as exempt.

IDAU_REGION183.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region183.

IDAU_REGION183.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region184.

IDAU_REGION184.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region184 as exempt.

IDAU_REGION184.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region184.

IDAU_REGION184.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region185.

IDAU_REGION185.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region185 as exempt.

IDAU_REGION185.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region185.

IDAU_REGION185.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region186.

IDAU_REGION186.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region186 as exempt.

IDAU_REGION186.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region186.

IDAU_REGION186.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region187.

IDAU_REGION187.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region187 as exempt.

IDAU_REGION187.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region187.

IDAU_REGION187.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region188.

IDAU_REGION188.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region188 as exempt.

IDAU_REGION188.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region188.

IDAU_REGION188.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region189.

IDAU_REGION189.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region189 as exempt.

IDAU_REGION189.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region189.

IDAU_REGION189.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region19.

IDAU_REGION19.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region19 as exempt.

IDAU_REGION19.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region19.

IDAU_REGION19.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region190.

IDAU_REGION190.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region190 as exempt.

IDAU_REGION190.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region190.

IDAU_REGION190.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region191.

IDAU_REGION191.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region191 as exempt.

IDAU_REGION191.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region191.

IDAU_REGION191.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region192.

IDAU_REGION192.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region192 as exempt.

IDAU_REGION192.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region192.

IDAU_REGION192.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region193.

IDAU_REGION193.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region193 as exempt.

IDAU_REGION193.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region193.

IDAU_REGION193.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region194.

IDAU_REGION194.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region194 as exempt.

IDAU_REGION194.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region194.

IDAU_REGION194.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region195.

IDAU_REGION195.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region195 as exempt.

IDAU_REGION195.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region195.

IDAU_REGION195.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region196.

IDAU_REGION196.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region196 as exempt.

IDAU_REGION196.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region196.

IDAU_REGION196.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region197.

IDAU_REGION197.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region197 as exempt.

IDAU_REGION197.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region197.

IDAU_REGION197.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region198.

IDAU_REGION198.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region198 as exempt.

IDAU_REGION198.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region198.

IDAU_REGION198.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region199.

IDAU_REGION199.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region199 as exempt.

IDAU_REGION199.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region199.

IDAU_REGION199.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region2.

IDAU_REGION2.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region2 as exempt.

IDAU_REGION2.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region2.

IDAU_REGION2.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region20.

IDAU_REGION20.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region20 as exempt.

IDAU_REGION20.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region20.

IDAU_REGION20.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region200.

IDAU_REGION200.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region200 as exempt.

IDAU_REGION200.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region200.

IDAU_REGION200.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region201.

IDAU_REGION201.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region201 as exempt.

IDAU_REGION201.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region201.

IDAU_REGION201.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region202.

IDAU_REGION202.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region202 as exempt.

IDAU_REGION202.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region202.

IDAU_REGION202.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region203.

IDAU_REGION203.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region203 as exempt.

IDAU_REGION203.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region203.

IDAU_REGION203.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region204.

IDAU_REGION204.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region204 as exempt.

IDAU_REGION204.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region204.

IDAU_REGION204.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region205.

IDAU_REGION205.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region205 as exempt.

IDAU_REGION205.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region205.

IDAU_REGION205.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region206.

IDAU_REGION206.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region206 as exempt.

IDAU_REGION206.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region206.

IDAU_REGION206.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region207.

IDAU_REGION207.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region207 as exempt.

IDAU_REGION207.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region207.

IDAU_REGION207.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region208.

IDAU_REGION208.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region208 as exempt.

IDAU_REGION208.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region208.

IDAU_REGION208.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region209.

IDAU_REGION209.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region209 as exempt.

IDAU_REGION209.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region209.

IDAU_REGION209.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region21.

IDAU_REGION21.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region21 as exempt.

IDAU_REGION21.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region21.

IDAU_REGION21.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region210.

IDAU_REGION210.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region210 as exempt.

IDAU_REGION210.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region210.

IDAU_REGION210.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region211.

IDAU_REGION211.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region211 as exempt.

IDAU_REGION211.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region211.

IDAU_REGION211.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region212.

IDAU_REGION212.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region212 as exempt.

IDAU_REGION212.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region212.

IDAU_REGION212.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region213.

IDAU_REGION213.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region213 as exempt.

IDAU_REGION213.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region213.

IDAU_REGION213.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region214.

IDAU_REGION214.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region214 as exempt.

IDAU_REGION214.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region214.

IDAU_REGION214.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region215.

IDAU_REGION215.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region215 as exempt.

IDAU_REGION215.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region215.

IDAU_REGION215.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region216.

IDAU_REGION216.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region216 as exempt.

IDAU_REGION216.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region216.

IDAU_REGION216.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region217.

IDAU_REGION217.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region217 as exempt.

IDAU_REGION217.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region217.

IDAU_REGION217.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region218.

IDAU_REGION218.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region218 as exempt.

IDAU_REGION218.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region218.

IDAU_REGION218.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region219.

IDAU_REGION219.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region219 as exempt.

IDAU_REGION219.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region219.

IDAU_REGION219.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region22.

IDAU_REGION22.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region22 as exempt.

IDAU_REGION22.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region22.

IDAU_REGION22.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region220.

IDAU_REGION220.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region220 as exempt.

IDAU_REGION220.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region220.

IDAU_REGION220.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region221.

IDAU_REGION221.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region221 as exempt.

IDAU_REGION221.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region221.

IDAU_REGION221.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region222.

IDAU_REGION222.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region222 as exempt.

IDAU_REGION222.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region222.

IDAU_REGION222.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region223.

IDAU_REGION223.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region223 as exempt.

IDAU_REGION223.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region223.

IDAU_REGION223.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region224.

IDAU_REGION224.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region224 as exempt.

IDAU_REGION224.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region224.

IDAU_REGION224.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region225.

IDAU_REGION225.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region225 as exempt.

IDAU_REGION225.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region225.

IDAU_REGION225.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region226.

IDAU_REGION226.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region226 as exempt.

IDAU_REGION226.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region226.

IDAU_REGION226.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region227.

IDAU_REGION227.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region227 as exempt.

IDAU_REGION227.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region227.

IDAU_REGION227.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region228.

IDAU_REGION228.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region228 as exempt.

IDAU_REGION228.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region228.

IDAU_REGION228.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region229.

IDAU_REGION229.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region229 as exempt.

IDAU_REGION229.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region229.

IDAU_REGION229.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region23.

IDAU_REGION23.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region23 as exempt.

IDAU_REGION23.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region23.

IDAU_REGION23.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region230.

IDAU_REGION230.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region230 as exempt.

IDAU_REGION230.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region230.

IDAU_REGION230.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region231.

IDAU_REGION231.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region231 as exempt.

IDAU_REGION231.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region231.

IDAU_REGION231.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region232.

IDAU_REGION232.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region232 as exempt.

IDAU_REGION232.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region232.

IDAU_REGION232.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region233.

IDAU_REGION233.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region233 as exempt.

IDAU_REGION233.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region233.

IDAU_REGION233.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region234.

IDAU_REGION234.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region234 as exempt.

IDAU_REGION234.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region234.

IDAU_REGION234.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region235.

IDAU_REGION235.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region235 as exempt.

IDAU_REGION235.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region235.

IDAU_REGION235.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region236.

IDAU_REGION236.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region236 as exempt.

IDAU_REGION236.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region236.

IDAU_REGION236.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region237.

IDAU_REGION237.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region237 as exempt.

IDAU_REGION237.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region237.

IDAU_REGION237.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region238.

IDAU_REGION238.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region238 as exempt.

IDAU_REGION238.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region238.

IDAU_REGION238.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region239.

IDAU_REGION239.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region239 as exempt.

IDAU_REGION239.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region239.

IDAU_REGION239.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region24.

IDAU_REGION24.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region24 as exempt.

IDAU_REGION24.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region24.

IDAU_REGION24.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region240.

IDAU_REGION240.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region240 as exempt.

IDAU_REGION240.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region240.

IDAU_REGION240.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region241.

IDAU_REGION241.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region241 as exempt.

IDAU_REGION241.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region241.

IDAU_REGION241.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region242.

IDAU_REGION242.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region242 as exempt.

IDAU_REGION242.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region242.

IDAU_REGION242.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region243.

IDAU_REGION243.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region243 as exempt.

IDAU_REGION243.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region243.

IDAU_REGION243.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region244.

IDAU_REGION244.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region244 as exempt.

IDAU_REGION244.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region244.

IDAU_REGION244.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region245.

IDAU_REGION245.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region245 as exempt.

IDAU_REGION245.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region245.

IDAU_REGION245.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region246.

IDAU_REGION246.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region246 as exempt.

IDAU_REGION246.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region246.

IDAU_REGION246.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region247.

IDAU_REGION247.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region247 as exempt.

IDAU_REGION247.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region247.

IDAU_REGION247.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region248.

IDAU_REGION248.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region248 as exempt.

IDAU_REGION248.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region248.

IDAU_REGION248.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region249.

IDAU_REGION249.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region249 as exempt.

IDAU_REGION249.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region249.

IDAU_REGION249.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region25.

IDAU_REGION25.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region25 as exempt.

IDAU_REGION25.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region25.

IDAU_REGION25.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region250.

IDAU_REGION250.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region250 as exempt.

IDAU_REGION250.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region250.

IDAU_REGION250.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region251.

IDAU_REGION251.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region251 as exempt.

IDAU_REGION251.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region251.

IDAU_REGION251.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region252.

IDAU_REGION252.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region252 as exempt.

IDAU_REGION252.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region252.

IDAU_REGION252.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region253.

IDAU_REGION253.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region253 as exempt.

IDAU_REGION253.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region253.

IDAU_REGION253.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region254.

IDAU_REGION254.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region254 as exempt.

IDAU_REGION254.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region254.

IDAU_REGION254.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region255.

IDAU_REGION255.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region255 as exempt.

IDAU_REGION255.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region255.

IDAU_REGION255.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region26.

IDAU_REGION26.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region26 as exempt.

IDAU_REGION26.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region26.

IDAU_REGION26.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region27.

IDAU_REGION27.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region27 as exempt.

IDAU_REGION27.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region27.

IDAU_REGION27.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region28.

IDAU_REGION28.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region28 as exempt.

IDAU_REGION28.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region28.

IDAU_REGION28.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region29.

IDAU_REGION29.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region29 as exempt.

IDAU_REGION29.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region29.

IDAU_REGION29.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region3.

IDAU_REGION3.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region3 as exempt.

IDAU_REGION3.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region3.

IDAU_REGION3.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region30.

IDAU_REGION30.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region30 as exempt.

IDAU_REGION30.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region30.

IDAU_REGION30.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region31.

IDAU_REGION31.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region31 as exempt.

IDAU_REGION31.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region31.

IDAU_REGION31.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region32.

IDAU_REGION32.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region32 as exempt.

IDAU_REGION32.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region32.

IDAU_REGION32.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region33.

IDAU_REGION33.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region33 as exempt.

IDAU_REGION33.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region33.

IDAU_REGION33.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region34.

IDAU_REGION34.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region34 as exempt.

IDAU_REGION34.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region34.

IDAU_REGION34.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region35.

IDAU_REGION35.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region35 as exempt.

IDAU_REGION35.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region35.

IDAU_REGION35.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region36.

IDAU_REGION36.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region36 as exempt.

IDAU_REGION36.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region36.

IDAU_REGION36.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region37.

IDAU_REGION37.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region37 as exempt.

IDAU_REGION37.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region37.

IDAU_REGION37.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region38.

IDAU_REGION38.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region38 as exempt.

IDAU_REGION38.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region38.

IDAU_REGION38.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region39.

IDAU_REGION39.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region39 as exempt.

IDAU_REGION39.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region39.

IDAU_REGION39.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region4.

IDAU_REGION4.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region4 as exempt.

IDAU_REGION4.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region4.

IDAU_REGION4.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region40.

IDAU_REGION40.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region40 as exempt.

IDAU_REGION40.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region40.

IDAU_REGION40.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region41.

IDAU_REGION41.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region41 as exempt.

IDAU_REGION41.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region41.

IDAU_REGION41.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region42.

IDAU_REGION42.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region42 as exempt.

IDAU_REGION42.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region42.

IDAU_REGION42.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region43.

IDAU_REGION43.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region43 as exempt.

IDAU_REGION43.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region43.

IDAU_REGION43.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region44.

IDAU_REGION44.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region44 as exempt.

IDAU_REGION44.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region44.

IDAU_REGION44.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region45.

IDAU_REGION45.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region45 as exempt.

IDAU_REGION45.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region45.

IDAU_REGION45.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region46.

IDAU_REGION46.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region46 as exempt.

IDAU_REGION46.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region46.

IDAU_REGION46.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region47.

IDAU_REGION47.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region47 as exempt.

IDAU_REGION47.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region47.

IDAU_REGION47.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region48.

IDAU_REGION48.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region48 as exempt.

IDAU_REGION48.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region48.

IDAU_REGION48.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region49.

IDAU_REGION49.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region49 as exempt.

IDAU_REGION49.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region49.

IDAU_REGION49.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region5.

IDAU_REGION5.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region5 as exempt.

IDAU_REGION5.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region5.

IDAU_REGION5.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region50.

IDAU_REGION50.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region50 as exempt.

IDAU_REGION50.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region50.

IDAU_REGION50.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region51.

IDAU_REGION51.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region51 as exempt.

IDAU_REGION51.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region51.

IDAU_REGION51.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region52.

IDAU_REGION52.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region52 as exempt.

IDAU_REGION52.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region52.

IDAU_REGION52.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region53.

IDAU_REGION53.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region53 as exempt.

IDAU_REGION53.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region53.

IDAU_REGION53.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region54.

IDAU_REGION54.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region54 as exempt.

IDAU_REGION54.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region54.

IDAU_REGION54.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region55.

IDAU_REGION55.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region55 as exempt.

IDAU_REGION55.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region55.

IDAU_REGION55.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region56.

IDAU_REGION56.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region56 as exempt.

IDAU_REGION56.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region56.

IDAU_REGION56.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region57.

IDAU_REGION57.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region57 as exempt.

IDAU_REGION57.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region57.

IDAU_REGION57.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region58.

IDAU_REGION58.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region58 as exempt.

IDAU_REGION58.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region58.

IDAU_REGION58.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region59.

IDAU_REGION59.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region59 as exempt.

IDAU_REGION59.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region59.

IDAU_REGION59.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region6.

IDAU_REGION6.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region6 as exempt.

IDAU_REGION6.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region6.

IDAU_REGION6.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region60.

IDAU_REGION60.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region60 as exempt.

IDAU_REGION60.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region60.

IDAU_REGION60.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region61.

IDAU_REGION61.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region61 as exempt.

IDAU_REGION61.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region61.

IDAU_REGION61.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region62.

IDAU_REGION62.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region62 as exempt.

IDAU_REGION62.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region62.

IDAU_REGION62.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region63.

IDAU_REGION63.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region63 as exempt.

IDAU_REGION63.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region63.

IDAU_REGION63.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region64.

IDAU_REGION64.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region64 as exempt.

IDAU_REGION64.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region64.

IDAU_REGION64.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region65.

IDAU_REGION65.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region65 as exempt.

IDAU_REGION65.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region65.

IDAU_REGION65.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region66.

IDAU_REGION66.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region66 as exempt.

IDAU_REGION66.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region66.

IDAU_REGION66.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region67.

IDAU_REGION67.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region67 as exempt.

IDAU_REGION67.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region67.

IDAU_REGION67.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region68.

IDAU_REGION68.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region68 as exempt.

IDAU_REGION68.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region68.

IDAU_REGION68.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region69.

IDAU_REGION69.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region69 as exempt.

IDAU_REGION69.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region69.

IDAU_REGION69.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region7.

IDAU_REGION7.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region7 as exempt.

IDAU_REGION7.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region7.

IDAU_REGION7.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region70.

IDAU_REGION70.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region70 as exempt.

IDAU_REGION70.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region70.

IDAU_REGION70.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region71.

IDAU_REGION71.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region71 as exempt.

IDAU_REGION71.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region71.

IDAU_REGION71.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region72.

IDAU_REGION72.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region72 as exempt.

IDAU_REGION72.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region72.

IDAU_REGION72.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region73.

IDAU_REGION73.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region73 as exempt.

IDAU_REGION73.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region73.

IDAU_REGION73.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region74.

IDAU_REGION74.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region74 as exempt.

IDAU_REGION74.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region74.

IDAU_REGION74.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region75.

IDAU_REGION75.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region75 as exempt.

IDAU_REGION75.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region75.

IDAU_REGION75.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region76.

IDAU_REGION76.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region76 as exempt.

IDAU_REGION76.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region76.

IDAU_REGION76.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region77.

IDAU_REGION77.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region77 as exempt.

IDAU_REGION77.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region77.

IDAU_REGION77.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region78.

IDAU_REGION78.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region78 as exempt.

IDAU_REGION78.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region78.

IDAU_REGION78.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region79.

IDAU_REGION79.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region79 as exempt.

IDAU_REGION79.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region79.

IDAU_REGION79.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region8.

IDAU_REGION8.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region8 as exempt.

IDAU_REGION8.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region8.

IDAU_REGION8.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region80.

IDAU_REGION80.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region80 as exempt.

IDAU_REGION80.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region80.

IDAU_REGION80.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region81.

IDAU_REGION81.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region81 as exempt.

IDAU_REGION81.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region81.

IDAU_REGION81.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region82.

IDAU_REGION82.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region82 as exempt.

IDAU_REGION82.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region82.

IDAU_REGION82.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region83.

IDAU_REGION83.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region83 as exempt.

IDAU_REGION83.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region83.

IDAU_REGION83.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region84.

IDAU_REGION84.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region84 as exempt.

IDAU_REGION84.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region84.

IDAU_REGION84.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region85.

IDAU_REGION85.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region85 as exempt.

IDAU_REGION85.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region85.

IDAU_REGION85.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region86.

IDAU_REGION86.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region86 as exempt.

IDAU_REGION86.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region86.

IDAU_REGION86.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region87.

IDAU_REGION87.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region87 as exempt.

IDAU_REGION87.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region87.

IDAU_REGION87.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region88.

IDAU_REGION88.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region88 as exempt.

IDAU_REGION88.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region88.

IDAU_REGION88.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region89.

IDAU_REGION89.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region89 as exempt.

IDAU_REGION89.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region89.

IDAU_REGION89.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region9.

IDAU_REGION9.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region9 as exempt.

IDAU_REGION9.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region9.

IDAU_REGION9.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region90.

IDAU_REGION90.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region90 as exempt.

IDAU_REGION90.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region90.

IDAU_REGION90.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region91.

IDAU_REGION91.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region91 as exempt.

IDAU_REGION91.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region91.

IDAU_REGION91.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region92.

IDAU_REGION92.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region92 as exempt.

IDAU_REGION92.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region92.

IDAU_REGION92.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region93.

IDAU_REGION93.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region93 as exempt.

IDAU_REGION93.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region93.

IDAU_REGION93.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region94.

IDAU_REGION94.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region94 as exempt.

IDAU_REGION94.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region94.

IDAU_REGION94.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region95.

IDAU_REGION95.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region95 as exempt.

IDAU_REGION95.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region95.

IDAU_REGION95.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region96.

IDAU_REGION96.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region96 as exempt.

IDAU_REGION96.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region96.

IDAU_REGION96.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region97.

IDAU_REGION97.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region97 as exempt.

IDAU_REGION97.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region97.

IDAU_REGION97.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region98.

IDAU_REGION98.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region98 as exempt.

IDAU_REGION98.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region98.

IDAU_REGION98.NSC**Type**

bool

Default value

0x0

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

Base address of IDAU region99.

IDAU_REGION99.ENABLE**Type**

bool

Default value

0x0

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

Mark IDAU region99 as exempt.

IDAU_REGION99.LADDR**Type**

int

Default value

0x0

Limit address of IDAU region99.

IDAU_REGION99.NSC

Type
bool

Default value
0x0

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

Number of IDAU regions.

3.10.46 IntegrityChecker

Integrity Checker. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1074: 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 23.

Iris and MTI instances for IntegrityChecker

This model has the following Iris instances:

Table 3-1075: IntegrityChecker Iris instances

InstanceName	ComponentName
IntegrityChecker	IntegrityChecker
IntegrityChecker.apb	PVBUSSlave

This model has the following MTI trace components:

Table 3-1076: IntegrityChecker MTI instances

InstanceName	ComponentName
IntegrityChecker.apb	PVBUSSlave

IntegrityChecker contains the following CADI targets:

- IntegrityChecker

Ports for IntegrityChecker

Table 3-1077: 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

ICBC Registers Reset Value.

ICDL_CHUNK_SIZE

Type

int

Default value

0x8

ICDL Chunk Size.

PID0_RESET_VALUE

Type

int

Default value

0x0

PID0 Registers Reset Value.

PID1_RESET_VALUE**Type**

int

Default value

0xb0

PID1 Registers Reset Value.

diagnostics**Type**

int

Default value

0x2

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.10.47 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 3-1078: 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 23.

Iris and MTI instances for IoTSS3_SystemControl

This model has the following Iris instances:

Table 3-1079: IoTSS3_SystemControl Iris instances

InstanceName	ComponentName
IoTSS3_SystemControl	IoTSS3_SystemControl
IoTSS3_SystemControl.PVBusMaster	PVBusMaster
IoTSS3_SystemControl.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-1080: IoTSS3_SystemControl MTI instances

InstanceName	ComponentName
IoTSS3_SystemControl.PVBusMaster	PVBusMaster
IoTSS3_SystemControl.PVBusSlave	PVBusSlave

IoTSS3_SystemControl contains the following CADI targets:

- IoTSS3_SystemControl

Ports for IoTSS3_SystemControl

Table 3-1081: 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	-
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_pvbus_m	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	-

Name	Protocol	Type	Description
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

Reset int32_t for INITSVTOR. Should match cpu<i>.INITSVTOR.

NUMCPU

Type

int

Default value

0x1

Number of Cortex-M CPU cores in the subsystem.

NUMVMBANK

Type

int

Default value

0x2

Number of Volatile Memory Banks.

SWRESETREQ_BIT

Type

int

Default value

0x9

Bit used for SWRESETREQ.

cpu0wait**Type**

bool

Default value

0x0

Whether to hold cpu0 in reset at boot.

cpu1wait**Type**

bool

Default value

0x1

Whether to hold cpu1 in reset at boot.

cpu2wait**Type**

bool

Default value

0x1

Whether to hold cpu2 in reset at boot.

cpu3wait**Type**

bool

Default value

0x1

Whether to hold cpu3 in reset at boot.

diagnostics**Type**

int

Default value

0x2

Diagnostics.

3.10.48 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 3-1082: 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 23.

Iris and MTI instances for IoTSS_AccessControlGate

This model has the following Iris instances:

Table 3-1083: IoTSS_AccessControlGate Iris instances

InstanceName	ComponentName
IoTSS_AccessControlGate	IoTSS_AccessControlGate
IoTSS_AccessControlGate.PVBusModifier	PVBusMapper

This model has the following MTI trace components:

Table 3-1084: IoTSS_AccessControlGate MTI instances

InstanceName	ComponentName
IoTSS_AccessControlGate.PVBusModifier	PVBusMapper

IoTSS_AccessControlGate contains the following CADI targets:

- IoTSS_AccessControlGate

Ports for IoTSS_AccessControlGate

Table 3-1085: Ports

Name	Protocol	Type	Description
ext_gate	Signal	Slave	-
master_ppuhwstat	Value	Slave	-
pvbus_m	PVBus	Master	-
pvbus_s	PVBus	Slave	-
slave_ppuhwstat	Value	Slave	-
wake_request	Signal	Master	-

Parameters for IoTSS_AccessControlGate

enabled

Type

bool

Default value

0x1

Enable the ACG. If disabled, will let all transactions through without side effects.

3.10.49 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 3-1086: 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 23.

Iris and MTI instances for IoTSS_PeripheralProtectionController

This model has the following Iris instances:

Table 3-1087: IoTSS_PeripheralProtectionController Iris instances

InstanceName	ComponentName
IoTSS_PeripheralProtectionController	IoTSS_PeripheralProtectionController
IoTSS_PeripheralProtectionController.PVBusModifier	PVBusMapper

This model has the following MTI trace components:

Table 3-1088: IoTSS_PeripheralProtectionController MTI instances

InstanceName	ComponentName
IoTSS_PeripheralProtectionController.PVBusModifier	PVBusMapper

IoTSS_PeripheralProtectionController contains the following CADI targets:

- IoTSS_PeripheralProtectionController

Ports for IoTSS_PeripheralProtectionController

Table 3-1089: Ports

Name	Protocol	Type	Description
cfg_ap	ValueState	Slave	-

Name	Protocol	Type	Description
cfg_nonsec	ValueState	Slave	-
cfg_sec_resp	ValueState	Slave	-
idau_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

Disable Memory gating logic.

NONSEC_MASK

Type

int

Default value

0x0

16-bit wide mask for security checking of ports: 0 = check, 1 = mask.

PORTx_ENABLE

Type

int

Default value

0xffff

Enable (1) or disable (0) port x (where x is between 0-15): enable = 1, disable = 0.

diagnostics

Type

int

Default value

0x0

Diagnostics.

3.10.50 KeyManagementUnit

KeyManagementUnit. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1090: 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 23.

Iris and MTI instances for KeyManagementUnit

This model has the following Iris instances:

Table 3-1091: KeyManagementUnit Iris instances

InstanceName	ComponentName
KeyManagementUnit	KeyManagementUnit
KeyManagementUnit.apb	PVBusSlave
KeyManagementUnit.keys_in	PVBusSlave

This model has the following MTI trace components:

Table 3-1092: KeyManagementUnit MTI instances

InstanceName	ComponentName
KeyManagementUnit.apb	PVBusSlave
KeyManagementUnit.keys_in	PVBusSlave

KeyManagementUnit contains the following CADI targets:

- KeyManagementUnit

Ports for KeyManagementUnit

Table 3-1093: Ports

Name	Protocol	Type	Description
apb	PVBus	Slave	register access via apb port
hw_keys_in	PVBus	Slave	HW key register access via private apb port - connect to LCM
irq_out	Signal	Master	IRQ signal out
keys_out	PVBus	Master	output keys via port
reset_in	Signal	Slave	Reset signal in

Parameters for KeyManagementUnit

KMUDKPARV0

Type

int

Default value

0x0

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

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

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

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

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

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

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

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.

KMUKSCRV0**Type**

int

Default value

0x0

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

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

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

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

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

Reset value of the KMUKSC<5> register. If no hardware key slot is supported (KMUNHWKSLTS is 5), this configuration option must be set to 0x0000_0000.

KMUKSCRV6**Type**

int

Default value

0x0

Reset value of the KMUKSC<6> register. If no hardware key slot is supported (KMUNHWKSLTS is 6), this configuration option must be set to 0x0000_0000.

KMUKSCRV7**Type**

int

Default value

0x0

Reset value of the KMUKSC<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

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

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

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

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.10.51 LifeCycleManager

LifeCycleManager. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1094: 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 23.

Iris and MTI instances for LifeCycleManager

This model has the following Iris instances:

Table 3-1095: 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 3-1096: LifeCycleManager MTI instances

InstanceName	ComponentName
LifeCycleManager.apb	PVBusSlave
LifeCycleManager.hw_keys_out	PVBusMaster
LifeCycleManager.nvm_external_out	PVBusMaster

LifeCycleManager contains the following CADI targets:

- RSE_LifeCycleManager

Ports for LifeCycleManager

Table 3-1097: Ports

Name	Protocol	Type	Description
apb	PVBus	Slave	APB4 Subordinate Interface - Access to LCM registers and internal OTP/NVM
hw_keys_out	PVBus	Master	APB manager interface for output of the HW Root of Trust keys
lcs	Value	Master	The 3-bit value of the LCS register
nvm_external_out	PVBus	Master	APB manager interface for external NVM (OTP)
reset_in	Signal	Slave	LCM reset in
sp_reset_out	Signal	Master	LCM SP_RST_REQ signal out

Parameters for LifeCycleManager

DCU_SP_DISABLE_MASK_VAL

Type

string

Default value

"0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

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

When set to 1, the Direct Key APB masking feature should be disabled.

KRTL_VAL

Type

string

Default value

0x7bfee0b730dcc241938458dafc2bc784cc15bdabeb84311f59cca8a9a4e8bc35

The Krtl value. (256-bit hex).

OTP_ADDR_WIDTH

Type

int

Default value

0xc

The OTP bus width (width in bits = OTP_ADDR_WIDTH + 2).

OTP_MASK_VAL**Type**

string

Default value

0x3aaf46ac792c53b0ffd80fe92129f0e8d80a9048799c5498c19f1a1ce7ddf5b20e2cc8f9456cf4ade2fe

This value must be generated using a true random number generator (1536-bit hex).

OTP_SIZE_IN_BYTES**Type**

int

Default value

0x4000

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"

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

"0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

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

"0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

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`"0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"`

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`"0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"`

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`"0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"`

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`"0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"`

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"`

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

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

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

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

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

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

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"

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"

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"

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"

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"

The default reset value of the 128-bit DCU signals when LCS=SE and TP Mode=TCI.

diagnostics

Type
int

Default value
0x2

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.10.52 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 3-1098: 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 23.

Iris and MTI instances for MMC

This model has the following Iris instances:

Table 3-1099: 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

MMC contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- MMC
- SchedulerThread
- SchedulerThreadEvent

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_STATUS_REG_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 3-1100: MMC registers

Name	CADI register number	Description
OCR_REG	0x000	Operating conditions register
CID_REG0	0x004	Card ID bits 127:96
CID_REG1	0x005	Card ID bits 95:64
CID_REG2	0x006	Card ID bits 63:32
CID_REG3	0x007	Card ID bits 31:0
CSD_REG0	0x008	Card specific data bits 127:96
CSD_REG1	0x009	Card specific data bits 95:64
CSD_REG2	0x00a	Card specific data bits 63:32
CSD_REG3	0x00b	Card specific data bit 31:0
RCA_REG	0x00c	Relative card address register
DSR_REG	0x00d	Driver stage register
BLOCKLEN_REG	0x00e	Block length
STATUS_REG	0x00f	Card status
BLOCK_COUNT_REG	0x010	Block count

Ports for MMC

Table 3-1101: 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"

Card type ('SD' or 'SDHC').

diagnostics

Type

int

Default value

0x0

Diagnostics level.

force_sector_addressing**Type**

bool

Default value

0x0

Use sector addressing even on small cards.

p_OEMid**Type**

int

Default value

0x0

Card ID OEM ID.

p_fast_access**Type**

bool

Default value

0x1

Don't simulate MMC block access delays.

p_manid**Type**

int

Default value

0x2

Card ID Manufacturer ID.

p_max_block_count**Type**

int

Default value

0x80

Default maximum block count reg. Default 0x80.

p_mmc_file**Type**

string

Default value

"mmc.dat"

MMCard filename.

p_prodName**Type**

string

Default value

"ARMmmc"

Card ID Product Name (6 chars).

p_prodRev**Type**

int

Default value

0x1

Card ID Product Revision.

p_sernum**Type**

int

Default value

0xca4d0001

Card Serial Number.

support_unpadded_images**Type**

bool

Default value

0x0

Support images that are not a multiple of 512k by padding them to the needed size (SDHC cards only).

3.10.53 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 3-1102: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for MMU_400

This model has the following Iris instances:

Table 3-1103: 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 3-1104: 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

MMU_400 contains the following CADI targets:

- MMU_400

- MMU_400_BASE

About MMU_400

Set the `use_label_mapping` parameter 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`. 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.

Ports for MMU_400

Table 3-1105: 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.
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.

Name	Protocol	Type	Description
cfg_flt_irpt_ns	Signal	Master	Non-secure configuration access fault interrupt. Corresponds to SMMU architectural signal SMMU_NSgCflrpt.
cfg_flt_irpt_s	Signal	Master	Secure configuration access fault interrupt. Corresponds to SMMU architectural signal SMMU_gCflrpt.
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.
glbl_flt_irpt_ns	Signal	Master	Global Non-secure fault interrupt. Corresponds to SMMU architectural signal SMMU_NSglrpt.
glbl_flt_irpt_s	Signal	Master	Global Secure fault interrupt. Corresponds to SMMU architectural signal SMMU_glrlpt.
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

""

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

cfg_cdtw

Type

bool

Default value

0x1

Perform coherent page table walks.

dump_unpredictability_in_user_flags

Type

bool

Default value

0x0

Override the user flags to encode unpredictable information (validation only).

label0_read_ssd**Type**

int

Default value

0x0

Label0: Read SDD or SSD_Index.

label0_read_stream_id**Type**

int

Default value

0x0

Label0: Read Stream ID.

label0_write_ssd**Type**

int

Default value

0x0

Label0: Write SDD or SSD_Index.

label0_write_stream_id**Type**

int

Default value

0x0

Label0: Write Stream ID.

label10_read_ssd**Type**

int

Default value

0x0

Label10: Read SDD or SSD_Index.

label10_read_stream_id**Type**

int

Default value

0x0

Label10: Read Stream ID.

label10_write_ssd**Type**

int

Default value

0x0

Label10: Write SDD or SSD_Index.

label10_write_stream_id**Type**

int

Default value

0x0

Label10: Write Stream ID.

label11_read_ssd**Type**

int

Default value

0x0

Label11: Read SDD or SSD_Index.

label11_read_stream_id**Type**

int

Default value

0x0

Label11: Read Stream ID.

label11_write_ssd**Type**

int

Default value

0x0

Label11: Write SDD or SSD_Index.

label11_write_stream_id**Type**

int

Default value

0x0

Label11: Write Stream ID.

label12_read_ssd**Type**

int

Default value

0x0

Label12: Read SDD or SSD_Index.

label12_read_stream_id**Type**

int

Default value

0x0

Label12: Read Stream ID.

label12_write_ssd**Type**

int

Default value

0x0

Label12: Write SDD or SSD_Index.

label12_write_stream_id**Type**

int

Default value

0x0

Label12: Write Stream ID.

label13_read_ssd**Type**

int

Default value

0x0

Label13: Read SDD or SSD_Index.

label13_read_stream_id**Type**

int

Default value

0x0

Label13: Read Stream ID.

label13_write_ssd**Type**

int

Default value

0x0

Label13: Write SDD or SSD_Index.

label13_write_stream_id**Type**

int

Default value

0x0

Label13: Write Stream ID.

label14_read_ssd**Type**

int

Default value

0x0

Label14: Read SDD or SSD_Index.

label14_read_stream_id**Type**

int

Default value

0x0

Label14: Read Stream ID.

label14_write_ssd**Type**

int

Default value

0x0

Label14: Write SDD or SSD_Index.

label14_write_stream_id**Type**

int

Default value

0x0

Label14: Write Stream ID.

label15_read_ssd**Type**

int

Default value

0x0

Label15: Read SDD or SSD_Index.

label15_read_stream_id**Type**

int

Default value

0x0

Label15: Read Stream ID.

label15_write_ssd**Type**

int

Default value

0x0

Label15: Write SDD or SSD_Index.

label15_write_stream_id**Type**

int

Default value

0x0

Label15: Write Stream ID.

label16_read_ssd**Type**

int

Default value

0x0

Label16: Read SDD or SSD_Index.

label16_read_stream_id**Type**

int

Default value

0x0

Label16: Read Stream ID.

label16_write_ssd**Type**

int

Default value

0x0

Label16: Write SDD or SSD_Index.

label16_write_stream_id**Type**

int

Default value

0x0

Label16: Write Stream ID.

label17_read_ssd**Type**

int

Default value

0x0

Label17: Read SDD or SSD_Index.

label17_read_stream_id**Type**

int

Default value

0x0

Label17: Read Stream ID.

label17_write_ssd**Type**

int

Default value

0x0

Label17: Write SDD or SSD_Index.

label17_write_stream_id**Type**

int

Default value

0x0

Label17: Write Stream ID.

label18_read_ssd**Type**

int

Default value

0x0

Label18: Read SDD or SSD_Index.

label18_read_stream_id**Type**

int

Default value

0x0

Label18: Read Stream ID.

label18_write_ssd**Type**

int

Default value

0x0

Label18: Write SDD or SSD_Index.

label18_write_stream_id**Type**

int

Default value

0x0

Label18: Write Stream ID.

label19_read_ssd**Type**

int

Default value

0x0

Label19: Read SDD or SSD_Index.

label19_read_stream_id**Type**

int

Default value

0x0

Label19: Read Stream ID.

label19_write_ssd**Type**

int

Default value

0x0

Label19: Write SDD or SSD_Index.

label19_write_stream_id**Type**

int

Default value

0x0

Label19: Write Stream ID.

label1_read_ssd**Type**

int

Default value

0x0

Label1: Read SDD or SSD_Index.

label1_read_stream_id**Type**

int

Default value

0x0

Label1: Read Stream ID.

label1_write_ssd**Type**

int

Default value

0x0

Label1: Write SDD or SSD_Index.

label1_write_stream_id**Type**

int

Default value

0x0

Label1: Write Stream ID.

label20_read_ssd**Type**

int

Default value

0x0

Label20: Read SDD or SSD_Index.

label20_read_stream_id**Type**

int

Default value

0x0

Label20: Read Stream ID.

label20_write_ssd**Type**

int

Default value

0x0

Label20: Write SDD or SSD_Index.

label20_write_stream_id**Type**

int

Default value

0x0

Label20: Write Stream ID.

label21_read_ssd**Type**

int

Default value

0x0

Label21: Read SDD or SSD_Index.

label21_read_stream_id**Type**

int

Default value

0x0

Label21: Read Stream ID.

label21_write_ssd**Type**

int

Default value

0x0

Label21: Write SDD or SSD_Index.

label21_write_stream_id**Type**

int

Default value

0x0

Label21: Write Stream ID.

label22_read_ssd**Type**

int

Default value

0x0

Label22: Read SDD or SSD_Index.

label22_read_stream_id**Type**

int

Default value

0x0

Label22: Read Stream ID.

label22_write_ssd**Type**

int

Default value

0x0

Label22: Write SDD or SSD_Index.

label22_write_stream_id**Type**

int

Default value

0x0

Label22: Write Stream ID.

label123_read_ssd**Type**

int

Default value

0x0

Label23: Read SDD or SSD_Index.

label123_read_stream_id**Type**

int

Default value

0x0

Label23: Read Stream ID.

label123_write_ssd**Type**

int

Default value

0x0

Label23: Write SDD or SSD_Index.

label123_write_stream_id**Type**

int

Default value

0x0

Label23: Write Stream ID.

label124_read_ssd**Type**

int

Default value

0x0

Label24: Read SDD or SSD_Index.

label24_read_stream_id**Type**

int

Default value

0x0

Label24: Read Stream ID.

label24_write_ssd**Type**

int

Default value

0x0

Label24: Write SDD or SSD_Index.

label24_write_stream_id**Type**

int

Default value

0x0

Label24: Write Stream ID.

label25_read_ssd**Type**

int

Default value

0x0

Label25: Read SDD or SSD_Index.

label25_read_stream_id**Type**

int

Default value

0x0

Label25: Read Stream ID.

label25_write_ssd**Type**

int

Default value

0x0

Label25: Write SDD or SSD_Index.

label25_write_stream_id**Type**

int

Default value

0x0

Label25: Write Stream ID.

label26_read_ssd**Type**

int

Default value

0x0

Label26: Read SDD or SSD_Index.

label26_read_stream_id**Type**

int

Default value

0x0

Label26: Read Stream ID.

label26_write_ssd**Type**

int

Default value

0x0

Label26: Write SDD or SSD_Index.

label26_write_stream_id**Type**

int

Default value

0x0

Label26: Write Stream ID.

label27_read_ssd**Type**

int

Default value

0x0

Label27: Read SDD or SSD_Index.

label27_read_stream_id**Type**

int

Default value

0x0

Label27: Read Stream ID.

label27_write_ssd**Type**

int

Default value

0x0

Label27: Write SDD or SSD_Index.

label27_write_stream_id**Type**

int

Default value

0x0

Label27: Write Stream ID.

label128_read_ssd**Type**

int

Default value

0x0

Label28: Read SDD or SSD_Index.

label128_read_stream_id**Type**

int

Default value

0x0

Label28: Read Stream ID.

label128_write_ssd**Type**

int

Default value

0x0

Label28: Write SDD or SSD_Index.

label128_write_stream_id**Type**

int

Default value

0x0

Label28: Write Stream ID.

label129_read_ssd**Type**

int

Default value

0x0

Label29: Read SDD or SSD_Index.

label29_read_stream_id**Type**

int

Default value

0x0

Label29: Read Stream ID.

label29_write_ssd**Type**

int

Default value

0x0

Label29: Write SDD or SSD_Index.

label29_write_stream_id**Type**

int

Default value

0x0

Label29: Write Stream ID.

label12_read_ssd**Type**

int

Default value

0x0

Label12: Read SDD or SSD_Index.

label12_read_stream_id**Type**

int

Default value

0x0

Label12: Read Stream ID.

label12_write_ssd**Type**

int

Default value

0x0

Label12: Write SDD or SSD_Index.

label12_write_stream_id**Type**

int

Default value

0x0

Label12: Write Stream ID.

label130_read_ssd**Type**

int

Default value

0x0

Label130: Read SDD or SSD_Index.

label130_read_stream_id**Type**

int

Default value

0x0

Label130: Read Stream ID.

label130_write_ssd**Type**

int

Default value

0x0

Label130: Write SDD or SSD_Index.

label30_write_stream_id**Type**

int

Default value

0x0

Label30: Write Stream ID.

label31_read_ssd**Type**

int

Default value

0x0

Label31: Read SDD or SSD_Index.

label31_read_stream_id**Type**

int

Default value

0x0

Label31: Read Stream ID.

label31_write_ssd**Type**

int

Default value

0x0

Label31: Write SDD or SSD_Index.

label31_write_stream_id**Type**

int

Default value

0x0

Label31: Write Stream ID.

label13_read_ssd**Type**

int

Default value

0x0

Label3: Read SDD or SSD_Index.

label13_read_stream_id**Type**

int

Default value

0x0

Label3: Read Stream ID.

label13_write_ssd**Type**

int

Default value

0x0

Label3: Write SDD or SSD_Index.

label13_write_stream_id**Type**

int

Default value

0x0

Label3: Write Stream ID.

label14_read_ssd**Type**

int

Default value

0x0

Label4: Read SDD or SSD_Index.

label4_read_stream_id**Type**

int

Default value

0x0

Label4: Read Stream ID.

label4_write_ssd**Type**

int

Default value

0x0

Label4: Write SDD or SSD_Index.

label4_write_stream_id**Type**

int

Default value

0x0

Label4: Write Stream ID.

label5_read_ssd**Type**

int

Default value

0x0

Label5: Read SDD or SSD_Index.

label5_read_stream_id**Type**

int

Default value

0x0

Label5: Read Stream ID.

label15_write_ssd**Type**

int

Default value

0x0

Label15: Write SDD or SSD_Index.

label15_write_stream_id**Type**

int

Default value

0x0

Label15: Write Stream ID.

label16_read_ssd**Type**

int

Default value

0x0

Label16: Read SDD or SSD_Index.

label16_read_stream_id**Type**

int

Default value

0x0

Label16: Read Stream ID.

label16_write_ssd**Type**

int

Default value

0x0

Label16: Write SDD or SSD_Index.

label6_write_stream_id**Type**

int

Default value

0x0

Label6: Write Stream ID.

label7_read_ssd**Type**

int

Default value

0x0

Label7: Read SDD or SSD_Index.

label7_read_stream_id**Type**

int

Default value

0x0

Label7: Read Stream ID.

label7_write_ssd**Type**

int

Default value

0x0

Label7: Write SDD or SSD_Index.

label7_write_stream_id**Type**

int

Default value

0x0

Label7: Write Stream ID.

label18_read_ssd**Type**

int

Default value

0x0

Label18: Read SDD or SSD_Index.

label18_read_stream_id**Type**

int

Default value

0x0

Label18: Read Stream ID.

label18_write_ssd**Type**

int

Default value

0x0

Label18: Write SDD or SSD_Index.

label18_write_stream_id**Type**

int

Default value

0x0

Label18: Write Stream ID.

label19_read_ssd**Type**

int

Default value

0x0

Label19: Read SDD or SSD_Index.

label9_read_stream_id**Type**

int

Default value

0x0

Label9: Read Stream ID.

label9_write_ssd**Type**

int

Default value

0x0

Label9: Write SDD or SSD_Index.

label9_write_stream_id**Type**

int

Default value

0x0

Label9: Write Stream ID.

mmu.seed**Type**

int

Default value

0x12345678

Seed for SMMU.

number_of_contexts**Type**

int

Default value

0x8

Number of context banks.

number_of_smrs**Type**

int

Default value

0x20

Number of stream match registers.

percent_tlbstatus_commits**Type**

int

Default value

0xa

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

prefetch_only_requests**Type**

int

Default value

0x0

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

""

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

""

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

ptw_has_separate_port**Type**

bool

Default value

0x1

Page Table Walks use pvbus_ptw_m.

pvbus_m_is_ace_lite**Type**

bool

Default value

0x1

Is pvbus_m (the downstream port that translated transaction exit) ACE-Lite.

pvbus_ptw_m_is_ace_lite**Type**

bool

Default value

0x1

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

StreamID bit width.

tlb_depth**Type**

int

Default value

0x40

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

use_label_mapping**Type**

bool

Default value

0x1

Use label mapping.

use_ssd_determination_table**Type**

bool

Default value

0x1

Use SSD Determination Table.

3.10.54 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 3-1106: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.**Iris and MTI instances for MMU_400_BASE**

This model has the following Iris instances:

Table 3-1107: 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 3-1108: 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

MMU_400_BASE contains the following CADI targets:

- MMU_400_BASE

Ports for MMU_400_BASE

Table 3-1109: 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.
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.

Name	Protocol	Type	Description
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

""

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

cfg_cttw

Type

bool

Default value

0x1

Perform coherent page table walks.

dump_unpredictability_in_user_flags

Type

bool

Default value

0x0

Override the user flags to encode unpredictable information (validation only).

number_of_contexts**Type**

int

Default value

0x8

Number of context banks.

number_of_smrs**Type**

int

Default value

0x10

Number of stream match registers.

percent_tlbstatus_commits**Type**

int

Default value

0xa

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

prefetch_only_requests**Type**

int

Default value

0x0

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

""

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

""

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

ptw_has_separate_port

Type

bool

Default value

0x1

Page Table Walks use pvbus_ptw_m.

pvbus_m_is_ace_lite

Type

bool

Default value

0x1

Is pvbus_m (the downstream port that translated transaction exit) ACE-Lite.

pvbus_ptw_m_is_ace_lite

Type

bool

Default value

0x1

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

Seed for SMMU.

stream_id_width

Type
int

Default value
0x6

StreamID bit width.

tlb_depth

Type
int

Default value
0x40

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

use_ssd_determination_table

Type
bool

Default value
0x1

Use SSD Determination Table.

3.10.55 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 3-1110: IP revisions support

Revision	Quality level
r2p4	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for MMU_500

This model has the following Iris instances:

Table 3-1111: MMU_500 Iris instances

InstanceName	ComponentName
MMU_500	MMU_500
MMU_500.mmu	MMU_500_BASE

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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

This model has the following MTI trace components:

Table 3-1112: 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

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

MMU_500 contains the following CADI targets:

- MMU_500
- MMU_500_BASE

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.

Ports for MMU_500

Table 3-1113: 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.
glblflt_irpt_ns	Signal	Master	Global Non-secure fault interrupt.
glblflt_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

""

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

cfg_cttw

Type

bool

Default value

0x1

Perform coherent page table walks.

dump_unpredictability_in_user_flags**Type**

bool

Default value

0x0

Override the user flags to encode unpredictable information (validation only).

label0_read_ssd**Type**

int

Default value

0x0

Label0: Read SDD or SSD_Index.

label0_read_stream_id**Type**

int

Default value

0x0

Label0: Read Stream ID.

label0_write_ssd**Type**

int

Default value

0x0

Label0: Write SDD or SSD_Index.

label0_write_stream_id**Type**

int

Default value

0x0

Label0: Write Stream ID.

label10_read_ssd**Type**

int

Default value

0x0

Label10: Read SDD or SSD_Index.

label10_read_stream_id**Type**

int

Default value

0x0

Label10: Read Stream ID.

label10_write_ssd**Type**

int

Default value

0x0

Label10: Write SDD or SSD_Index.

label10_write_stream_id**Type**

int

Default value

0x0

Label10: Write Stream ID.

label11_read_ssd**Type**

int

Default value

0x0

Label11: Read SDD or SSD_Index.

label11_read_stream_id**Type**

int

Default value

0x0

Label11: Read Stream ID.

label11_write_ssd**Type**

int

Default value

0x0

Label11: Write SDD or SSD_Index.

label11_write_stream_id**Type**

int

Default value

0x0

Label11: Write Stream ID.

label12_read_ssd**Type**

int

Default value

0x0

Label12: Read SDD or SSD_Index.

label12_read_stream_id**Type**

int

Default value

0x0

Label12: Read Stream ID.

label12_write_ssd**Type**

int

Default value

0x0

Label12: Write SDD or SSD_Index.

label12_write_stream_id**Type**

int

Default value

0x0

Label12: Write Stream ID.

label13_read_ssd**Type**

int

Default value

0x0

Label13: Read SDD or SSD_Index.

label13_read_stream_id**Type**

int

Default value

0x0

Label13: Read Stream ID.

label13_write_ssd**Type**

int

Default value

0x0

Label13: Write SDD or SSD_Index.

label13_write_stream_id**Type**

int

Default value

0x0

Label13: Write Stream ID.

label14_read_ssd**Type**

int

Default value

0x0

Label14: Read SDD or SSD_Index.

label14_read_stream_id**Type**

int

Default value

0x0

Label14: Read Stream ID.

label14_write_ssd**Type**

int

Default value

0x0

Label14: Write SDD or SSD_Index.

label14_write_stream_id**Type**

int

Default value

0x0

Label14: Write Stream ID.

label15_read_ssd**Type**

int

Default value

0x0

Label15: Read SDD or SSD_Index.

label15_read_stream_id**Type**

int

Default value

0x0

Label15: Read Stream ID.

label15_write_ssd**Type**

int

Default value

0x0

Label15: Write SDD or SSD_Index.

label15_write_stream_id**Type**

int

Default value

0x0

Label15: Write Stream ID.

label16_read_ssd**Type**

int

Default value

0x0

Label16: Read SDD or SSD_Index.

label16_read_stream_id**Type**

int

Default value

0x0

Label16: Read Stream ID.

label16_write_ssd**Type**

int

Default value

0x0

Label16: Write SDD or SSD_Index.

label16_write_stream_id**Type**

int

Default value

0x0

Label16: Write Stream ID.

label17_read_ssd**Type**

int

Default value

0x0

Label17: Read SDD or SSD_Index.

label17_read_stream_id**Type**

int

Default value

0x0

Label17: Read Stream ID.

label17_write_ssd**Type**

int

Default value

0x0

Label17: Write SDD or SSD_Index.

label17_write_stream_id**Type**

int

Default value

0x0

Label17: Write Stream ID.

label18_read_ssd**Type**

int

Default value

0x0

Label18: Read SDD or SSD_Index.

label18_read_stream_id**Type**

int

Default value

0x0

Label18: Read Stream ID.

label18_write_ssd**Type**

int

Default value

0x0

Label18: Write SDD or SSD_Index.

label18_write_stream_id**Type**

int

Default value

0x0

Label18: Write Stream ID.

label19_read_ssd**Type**

int

Default value

0x0

Label19: Read SDD or SSD_Index.

label19_read_stream_id**Type**

int

Default value

0x0

Label19: Read Stream ID.

label19_write_ssd**Type**

int

Default value

0x0

Label19: Write SDD or SSD_Index.

label19_write_stream_id**Type**

int

Default value

0x0

Label19: Write Stream ID.

label11_read_ssd**Type**

int

Default value

0x0

Label1: Read SDD or SSD_Index.

label11_read_stream_id**Type**

int

Default value

0x0

Label1: Read Stream ID.

label11_write_ssd**Type**

int

Default value

0x0

Label1: Write SDD or SSD_Index.

label11_write_stream_id**Type**

int

Default value

0x0

Label1: Write Stream ID.

label20_read_ssd**Type**

int

Default value

0x0

Label20: Read SDD or SSD_Index.

label20_read_stream_id**Type**

int

Default value

0x0

Label20: Read Stream ID.

label20_write_ssd**Type**

int

Default value

0x0

Label20: Write SDD or SSD_Index.

label20_write_stream_id**Type**

int

Default value

0x0

Label20: Write Stream ID.

label21_read_ssd**Type**

int

Default value

0x0

Label21: Read SDD or SSD_Index.

label21_read_stream_id**Type**

int

Default value

0x0

Label21: Read Stream ID.

label21_write_ssd**Type**

int

Default value

0x0

Label21: Write SDD or SSD_Index.

label21_write_stream_id**Type**

int

Default value

0x0

Label21: Write Stream ID.

label22_read_ssd**Type**

int

Default value

0x0

Label22: Read SDD or SSD_Index.

label22_read_stream_id**Type**

int

Default value

0x0

Label22: Read Stream ID.

label22_write_ssd**Type**

int

Default value

0x0

Label22: Write SDD or SSD_Index.

label22_write_stream_id**Type**

int

Default value

0x0

Label22: Write Stream ID.

label23_read_ssd**Type**

int

Default value

0x0

Label23: Read SDD or SSD_Index.

label23_read_stream_id**Type**

int

Default value

0x0

Label23: Read Stream ID.

label23_write_ssd**Type**

int

Default value

0x0

Label23: Write SDD or SSD_Index.

label23_write_stream_id**Type**

int

Default value

0x0

Label23: Write Stream ID.

label24_read_ssd**Type**

int

Default value

0x0

Label24: Read SDD or SSD_Index.

label24_read_stream_id**Type**

int

Default value

0x0

Label24: Read Stream ID.

label24_write_ssd**Type**

int

Default value

0x0

Label24: Write SDD or SSD_Index.

label24_write_stream_id**Type**

int

Default value

0x0

Label24: Write Stream ID.

label25_read_ssd**Type**

int

Default value

0x0

Label25: Read SDD or SSD_Index.

label125_read_stream_id**Type**

int

Default value

0x0

Label25: Read Stream ID.

label125_write_ssd**Type**

int

Default value

0x0

Label25: Write SDD or SSD_Index.

label125_write_stream_id**Type**

int

Default value

0x0

Label25: Write Stream ID.

label126_read_ssd**Type**

int

Default value

0x0

Label26: Read SDD or SSD_Index.

label126_read_stream_id**Type**

int

Default value

0x0

Label26: Read Stream ID.

label26_write_ssd**Type**

int

Default value

0x0

Label26: Write SDD or SSD_Index.

label26_write_stream_id**Type**

int

Default value

0x0

Label26: Write Stream ID.

label27_read_ssd**Type**

int

Default value

0x0

Label27: Read SDD or SSD_Index.

label27_read_stream_id**Type**

int

Default value

0x0

Label27: Read Stream ID.

label27_write_ssd**Type**

int

Default value

0x0

Label27: Write SDD or SSD_Index.

label27_write_stream_id**Type**

int

Default value

0x0

Label27: Write Stream ID.

label28_read_ssd**Type**

int

Default value

0x0

Label28: Read SDD or SSD_Index.

label28_read_stream_id**Type**

int

Default value

0x0

Label28: Read Stream ID.

label28_write_ssd**Type**

int

Default value

0x0

Label28: Write SDD or SSD_Index.

label28_write_stream_id**Type**

int

Default value

0x0

Label28: Write Stream ID.

label29_read_ssd**Type**

int

Default value

0x0

Label29: Read SDD or SSD_Index.

label29_read_stream_id**Type**

int

Default value

0x0

Label29: Read Stream ID.

label29_write_ssd**Type**

int

Default value

0x0

Label29: Write SDD or SSD_Index.

label29_write_stream_id**Type**

int

Default value

0x0

Label29: Write Stream ID.

label12_read_ssd**Type**

int

Default value

0x0

Label12: Read SDD or SSD_Index.

label12_read_stream_id**Type**

int

Default value

0x0

Label12: Read Stream ID.

label12_write_ssd**Type**

int

Default value

0x0

Label12: Write SDD or SSD_Index.

label12_write_stream_id**Type**

int

Default value

0x0

Label12: Write Stream ID.

label130_read_ssd**Type**

int

Default value

0x0

Label130: Read SDD or SSD_Index.

label130_read_stream_id**Type**

int

Default value

0x0

Label130: Read Stream ID.

label30_write_ssd**Type**

int

Default value

0x0

Label30: Write SDD or SSD_Index.

label30_write_stream_id**Type**

int

Default value

0x0

Label30: Write Stream ID.

label31_read_ssd**Type**

int

Default value

0x0

Label31: Read SDD or SSD_Index.

label31_read_stream_id**Type**

int

Default value

0x0

Label31: Read Stream ID.

label31_write_ssd**Type**

int

Default value

0x0

Label31: Write SDD or SSD_Index.

label31_write_stream_id**Type**

int

Default value

0x0

Label31: Write Stream ID.

label3_read_ssd**Type**

int

Default value

0x0

Label3: Read SDD or SSD_Index.

label3_read_stream_id**Type**

int

Default value

0x0

Label3: Read Stream ID.

label3_write_ssd**Type**

int

Default value

0x0

Label3: Write SDD or SSD_Index.

label3_write_stream_id**Type**

int

Default value

0x0

Label3: Write Stream ID.

label14_read_ssd**Type**

int

Default value

0x0

Label4: Read SDD or SSD_Index.

label14_read_stream_id**Type**

int

Default value

0x0

Label4: Read Stream ID.

label14_write_ssd**Type**

int

Default value

0x0

Label4: Write SDD or SSD_Index.

label14_write_stream_id**Type**

int

Default value

0x0

Label4: Write Stream ID.

label15_read_ssd**Type**

int

Default value

0x0

Label5: Read SDD or SSD_Index.

label15_read_stream_id**Type**

int

Default value

0x0

Label5: Read Stream ID.

label15_write_ssd**Type**

int

Default value

0x0

Label5: Write SDD or SSD_Index.

label15_write_stream_id**Type**

int

Default value

0x0

Label5: Write Stream ID.

label16_read_ssd**Type**

int

Default value

0x0

Label6: Read SDD or SSD_Index.

label16_read_stream_id**Type**

int

Default value

0x0

Label6: Read Stream ID.

label6_write_ssd**Type**

int

Default value

0x0

Label6: Write SDD or SSD_Index.

label6_write_stream_id**Type**

int

Default value

0x0

Label6: Write Stream ID.

label7_read_ssd**Type**

int

Default value

0x0

Label7: Read SDD or SSD_Index.

label7_read_stream_id**Type**

int

Default value

0x0

Label7: Read Stream ID.

label7_write_ssd**Type**

int

Default value

0x0

Label7: Write SDD or SSD_Index.

label17_write_stream_id**Type**

int

Default value

0x0

Label7: Write Stream ID.

label18_read_ssd**Type**

int

Default value

0x0

Label8: Read SDD or SSD_Index.

label18_read_stream_id**Type**

int

Default value

0x0

Label8: Read Stream ID.

label18_write_ssd**Type**

int

Default value

0x0

Label8: Write SDD or SSD_Index.

label18_write_stream_id**Type**

int

Default value

0x0

Label8: Write Stream ID.

label9_read_ssd**Type**

int

Default value

0x0

Label9: Read SDD or SSD_Index.

label9_read_stream_id**Type**

int

Default value

0x0

Label9: Read Stream ID.

label9_write_ssd**Type**

int

Default value

0x0

Label9: Write SDD or SSD_Index.

label9_write_stream_id**Type**

int

Default value

0x0

Label9: Write Stream ID.

mmu.PRIVATE_PARAMETER_personality**Type**

string

Default value

""

The personality to use (affects ID codes and various imp def features).

mmu.PRIVATE_PARAMETER_seed**Type**

int

Default value

0x12345678

Seed for randomised SMMU implementation defined behaviour.

mmu.PRIVATE_PARAMETER_validation_mode**Type**

int

Default value

0x0

Internal validation mode.

number_of_contexts**Type**

int

Default value

0x8

Number of context banks.

number_of_smrs**Type**

int

Default value

0x20

Number of stream match registers.

percent_tlbstatus_commits**Type**

int

Default value

0xa

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

prefetch_only_requests**Type**

int

Default value

0x0

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

""

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

""

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

ptw_has_separate_port**Type**

bool

Default value

0x1

Page Table Walks use pvbus_ptw_m.

supports_nested_translations**Type**

bool

Default value

0x1

Supports nested translations (stage 1 + stage 2).

tlb_depth

Type
int

Default value
0x800

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

use_label_mapping

Type
bool

Default value
0x1

Use label mapping.

use_ssd_determination_table

Type
bool

Default value
0x1

Use SSD Determination Table.

version

Type
string

Default value
"EAC"

Version of the RTL that the model represents. Valid values are LACr1 and EAC.

3.10.56 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 3-1114: IP revisions support

Revision	Quality level
r2p4	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for MMU_500_BASE

This model has the following Iris instances:

Table 3-1115: 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 3-1116: 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

MMU_500_BASE contains the following CADI targets:

- MMU_500_BASE

Ports for MMU_500_BASE

Table 3-1117: 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.
glbl_flt_irpt_ns	Signal	Master	Global non-secure Fault Interrupt In the SMMU Architecture this is called SMMU_NSglrpt.
glbl_flt_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

""

The personality to use (affects ID codes and various imp def features).

PRIVATE_PARAMETER_seed

Type

int

Default value

0x12345678

Seed for randomised SMMU implementation defined behaviour.

PRIVATE_PARAMETER_validation_mode

Type

int

Default value

0x0

Internal validation mode.

always_secure_ssd_indices**Type**

string

Default value

""

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

cfg_cttw**Type**

bool

Default value

0x1

Perform coherent page table walks.

dump_unpredictability_in_user_flags**Type**

bool

Default value

0x0

Override the user flags to encode unpredictable information (validation only).

number_of_contexts**Type**

int

Default value

0x8

Number of context banks.

number_of_smrs**Type**

int

Default value

0x10

Number of stream match registers.

percent_tlbstatus_commits**Type**

int

Default value

0xa

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

prefetch_only_requests**Type**

int

Default value

0x0

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

""

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

""

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

ptw_has_separate_port**Type**

bool

Default value

0x1

Page Table Walks use pvbus_ptw_m (or uses pvbus_m[0]).

supports_nested_translations

Type
bool

Default value
0x1

Supports nested translations (stage 1 + stage 2).

tlb_depth

Type
int

Default value
0x800

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

use_ssd_determination_table

Type
bool

Default value
0x1

Use SSD Determination Table.

version

Type
string

Default value
"EAC"

Version of the RTL that the model represents. Valid values are LACr1 and EAC.

3.10.57 MMU_600

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1118: IP revisions support

Revision	Quality level
r0p0	Full support
r0p1	Full support

Revision	Quality level
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 23.

Iris and MTI instances for MMU_600

This model has the following Iris instances:

Table 3-1119: 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
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.tb[46]	PVBusMapper
MMU_600.tb[47]	PVBusMapper
MMU_600.tb[48]	PVBusMapper
MMU_600.tb[49]	PVBusMapper
MMU_600.tb[4]	PVBusMapper
MMU_600.tb[50]	PVBusMapper
MMU_600.tb[51]	PVBusMapper
MMU_600.tb[52]	PVBusMapper
MMU_600.tb[53]	PVBusMapper
MMU_600.tb[54]	PVBusMapper
MMU_600.tb[55]	PVBusMapper
MMU_600.tb[56]	PVBusMapper
MMU_600.tb[57]	PVBusMapper
MMU_600.tb[58]	PVBusMapper
MMU_600.tb[59]	PVBusMapper
MMU_600.tb[5]	PVBusMapper
MMU_600.tb[60]	PVBusMapper
MMU_600.tb[61]	PVBusMapper
MMU_600.tb[62]	PVBusMapper
MMU_600.tb[63]	PVBusMapper
MMU_600.tb[6]	PVBusMapper
MMU_600.tb[7]	PVBusMapper
MMU_600.tb[8]	PVBusMapper
MMU_600.tb[9]	PVBusMapper

This model has the following MTI trace components:

Table 3-1120: 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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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
MMU_600.tbu[61]	PVBusMapper
MMU_600.tbu[62]	PVBusMapper
MMU_600.tbu[63]	PVBusMapper
MMU_600.tbu[6]	PVBusMapper
MMU_600.tbu[7]	PVBusMapper

InstanceName	ComponentName
MMU_600.tbu[8]	PVBusMapper
MMU_600.tbu[9]	PVBusMapper

MMU_600 contains the following CADI targets:

- MMU_600

Limitations

- PMU does not implement all architecturally-mandated events:
 - The PMU is intended for demonstration purposes only
 - Note: between 11.17 and 11.18 then 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.
- No RAS
- No power control
- No sup_oas that controls the OAS of the SMMU. This is expected to be constant for a system
- Cache maintenance operations cannot be inserted into the SMMU's TBU ports
- PVBus has no representation of the cache stash operations and so they are not supported.
- 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



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 then used for subsequent sufficiently similar transactions without needing the intervention of the SMMU model (and so won't be traced, etc.)

Notes

- 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 used will be listed in 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.
- A bad configuration will render the model inactive.
- Some configuration 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, 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.
- 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.
- 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 one.
- Debug and real accesses to the registers must be 32 or 64 bits.
- MSIs are issued on the `qtw_pvbus_m` port using attributes determined by the parameter `msi_attribute_transform`, whilst Event Queue writes will always be 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.
- 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 will be forced to iNC-oNC-osh. For example, if a NoSnoop transaction uses a page table and is transformed to iWB-oWB-nsh then it will be 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 will exit 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 are interpreted as ! NoSnoop
 - Thus normalized to iWB-oWB-ish
 - Anything else is considered NoSnoop
 - Thus 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 will replicate the outer attribute into the inner attribute and then perform the normalisation 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. The normalisation is:
 - If the transaction is classified as a PCIe one then the NoSnoop transform described above is applied. That is:
 - if the original transaction was NoSnoop then any weaker memory type is strengthened to iNC-oNC-osh, then 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.

Ports for MMU_600

Table 3-1121: 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
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
tbu_reset_in[62]	Signal	Slave	Reset signals The TBUs can have independent reset signals. Each signal <code>tbu_reset_in[n]</code> corresponds to the TBU using <code>tbs_pvbus_s[n]/tbs_pvbus_s[n]</code> pair. If the SMMU receives a transaction whilst the TBU is expected to be in reset then it will complain using the <code>ArchMsg.Warning.warning</code> trace source. Those <code>tbu_reset_in</code> 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 <code>TCU_NODE_STATUS</code> 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`

Copyright © 2017–2024 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- **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 `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]
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], ...
```

The `streamID` (32 b) is valid if `sidv` is 1 or both `sidv` and `nsidv` are unused.

`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].

The `substreamID` (20 b) is valid if `ssv` is true.

`sidv == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

`nSEC_SID`, `nSSV`, `nsidv` are available with negative logic. Different attributes are independent and can use negative or positive logic. 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 `NSID_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`.

`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 pvbush_id_routed_m port as DTI-ATS is bidirectional, but PVBush 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, MasterID=0xFFFFffffff”

Transform downstream attributes of MSI transactions.



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 was 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.

- "" or "none" – the input and output attributes are identical.
- How to alter the output attributes, for example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV,
UserFlags[19:0]=SubstreamID"
```

The attributes that can appear on the left hand side of the transform are `ExtendedID`, `MasterID` and `UserFlags`.

The source attributes that can be used are:

ExtendedID/MasterID/UserFlags

the incoming attributes.

DeviceID`StreamID + translated_device_id_base``StreamID/SubstreamID/SSV/SEC_SID`**nSEC_SID/nSSV**

the negative logic versions.

St1PBHA/St2PBHA

the Page Based Hardware Attributes from any used leaf descriptors (or zero if not used).

STE_IMPDEF1`STE[127:116]`

The right hand side may also contain numeric literals. Any bits of the attributes that have no transform specified are retained from the input.

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"

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/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 use 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.COACC`

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.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

- ExtendedID/MasterID/UserFlags

RHS Symbols:

- HWATTR_KIND_0: PBHA information
- 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

- numeric literals.

ExtendedID/MasterID/UserFlags start with values {0, 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

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_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 $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$.

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 $[0, \max(2, (t \ \& \ 0xFFFFffff)) - 1]$.

3.10.58 MMU_700

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1123: 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 23.

Iris and MTI instances for MMU_700

This model has the following Iris instances:

Table 3-1124: MMU_700 Iris instances

InstanceName	ComponentName
MMU_700	MMU_700
MMU_700.register_file[0]	PVBusSlave
MMU_700.service_request_tbu[0]	PVBusSlave

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
MMU_700.tbv[59]	PVBusMapper
MMU_700.tbv[5]	PVBusMapper
MMU_700.tbv[60]	PVBusMapper
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 3-1125: 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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
MMU_700.tbv[43]	PVBusMapper
MMU_700.tbv[44]	PVBusMapper
MMU_700.tbv[45]	PVBusMapper
MMU_700.tbv[46]	PVBusMapper
MMU_700.tbv[47]	PVBusMapper
MMU_700.tbv[48]	PVBusMapper
MMU_700.tbv[49]	PVBusMapper
MMU_700.tbv[4]	PVBusMapper
MMU_700.tbv[50]	PVBusMapper
MMU_700.tbv[51]	PVBusMapper
MMU_700.tbv[52]	PVBusMapper
MMU_700.tbv[53]	PVBusMapper
MMU_700.tbv[54]	PVBusMapper
MMU_700.tbv[55]	PVBusMapper
MMU_700.tbv[56]	PVBusMapper
MMU_700.tbv[57]	PVBusMapper
MMU_700.tbv[58]	PVBusMapper
MMU_700.tbv[59]	PVBusMapper
MMU_700.tbv[5]	PVBusMapper
MMU_700.tbv[60]	PVBusMapper
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

MMU_700 contains the following CADI targets:

- MMU_700

Limitations

- PMU does not implement all architecturally mandated events:
 - The PMU is intended for demonstration purposes only
 - Note: between 11.17 and 11.18 then the point of triggering for events changed for 'TLB miss' and this might lead to an (architecturally
- Limited RAS
- No power control

- The IMP DEF MPAM register file is not implemented. This controls how the internal resources of the MMU-700 are partitioned.
- No sup_oas that controls the OAS of the SMMU. This is expected to be constant for a system
- tcu_sid[31:0] is not modelled, instead the parameter `smmu_msi_device_id` is used.
- PVBUS has no representation of the cache stash operations and so they are not supported.
- The SYSCO interface is not implemented
- The low power interface is not implemented
- `TCU_STATUS.GNT_XLATE_SLOTS` always reads at 512
- 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 then 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 behaviour and the Update bit is **RAZ/WI**.



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 then used for subsequent sufficiently similar transactions without needing the intervention of the SMMU model (and so won't be traced, etc.)

Notes {section}

- 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 used will be listed in 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.
- A bad configuration will renders the model inactive.
- Some configuration 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 then the component driving them must also drive this transition again.
- 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.
- 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 one.
- Debug and real accesses to the registers must be 32 or 64 bits.
- MSIs are issued on the `qtw_pvbuss_m` port using attributes determined by the parameter `msi_attribute_transform`, whilst Event Queue writes will always be 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.
- 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 will be forced to iNC-oNC-osh. For example, if a NoSnoop transaction uses a page table and is transformed to iWB-oWB-nsh then it will be 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 will exit 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 are interpreted as ! NoSnoop
 - thus normalized to iWB-oWB-ish
 - anything else is considered NoSnoop
 - thus 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 will replicate the outer attribute into the inner attribute and then perform the normalisation 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. The normalisation is:

- If the transaction is classified as a PCIe one then the NoSnoop transform described above is applied. That is:
 - if the original transaction was NoSnoop then any weaker memory type is strengthened to iNC-oNC-osh, then 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.

Ports for MMU_700

Table 3-1126: 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_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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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. 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.
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.



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`.



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`.



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**`0xFFFFffffff`

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**

`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]
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], ...
```

The `streamID` (32 b) is valid if `sidv` is 1 or both `sidv` and `nsidv` are unused.

`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].

The `substreamID` (20 b) is valid if `ssv` is true.

`sidv == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

`nSEC_SID`, `nSSV`, `nSIDV` are available with negative logic. Different attributes are independent and can use negative or positive logic. 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`.

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.

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"

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/MasterID/UserFlags`.

RHS Symbols:

- `MPAM_PARTID`
- `MPAM_PMG`
- `MPAM_NS`
- numeric literals

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, MasterID=0xFFFFFFFF"

Transform downstream attributes of MSI transactions.



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 was 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.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV,
UserFlags[19:0]=SubstreamID,
MasterID[10]=MasterID[11], MasterID[11]=MasterID[10]"
```

RHS/LHS Symbols:

- ExtendedID/MasterID/UserFlags: incoming/outgoing attributes.

RHS Symbols:

- **DeviceID**
StreamID + translated_device_id_base
- StreamID/SubstreamID/SSV/SEC_SID
- **nSEC_SID/nSSV**
negative logic versions.
- **st1PBHA/st2PBHA**
Page Based Hardware Attributes from leaf descriptors (zero if unused).
- **STE_IMPDEF1**
STE[127:116]
- numeric literals.

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.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV, UserFlags[19:0]=SubstreamID,
MasterID[10]=MasterID[11], MasterID[11]=MasterID[10]"
```

RHS/LHS Symbols:

- **ExtendedID/MasterID/UserFlags**
incoming/outgoing attributes.

RHS Symbols:

- **DeviceID**
StreamID + translated_device_id_base
- StreamID/SubstreamID/SSV/SEC_SID
- **nSSV/nSEC_SID/**
negative logic versions.
- **St1PBHA/St2PBHA**
Page Based Hardware Attributes from leaf descriptors (zero if not used).
- **STE_IMPDEF1**
STE[127:116]
- **HWATTR_KIND_0**
PBHA information
- numeric literals.
- SIDV = 1, nSIDV = 0 (fixed values to indicate StreamID)

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_NoStreamID for NoStreamID transactions.

output_id_routed_transform

Type

string

Default value

“Address[27:12]=StreamID[15:0], PAS=SSD”

The SMMU generates the following ID-routed transaction on the pvb_{bus_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/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 use 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.COHAACC`

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.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

- ExtendedID/MasterID/UserFlags

RHS Symbols:

- HWATTR_KIND_0: PBHA information
- 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

- numeric literals.

ExtendedID/MasterID/UserFlags start with values {0, 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 $[0, \max(2, (t \& 0xFFFFffffff) - 1)]$.

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 $[0, \max(2, (t \& 0xFFFFffffff) - 1)]$.

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 $[0, \max(2, (t \& 0xFFFFffff)) - 1]$.

3.10.59 MMU_S3

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1128: 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 23.

Changes in 11.27.17**Table 3-1129: Model quality changes**

From	To
r0p0=rel	r0p0=rel
r1p0=pre	r1p0=rel

Iris and MTI instances for MMU_S3

This model has the following Iris instances:

Table 3-1130: 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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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.tbv[0]	PVBusMapper
MMU_S3.tbv[10]	PVBusMapper
MMU_S3.tbv[11]	PVBusMapper
MMU_S3.tbv[12]	PVBusMapper
MMU_S3.tbv[13]	PVBusMapper
MMU_S3.tbv[14]	PVBusMapper
MMU_S3.tbv[15]	PVBusMapper
MMU_S3.tbv[16]	PVBusMapper
MMU_S3.tbv[17]	PVBusMapper
MMU_S3.tbv[18]	PVBusMapper
MMU_S3.tbv[19]	PVBusMapper
MMU_S3.tbv[1]	PVBusMapper
MMU_S3.tbv[20]	PVBusMapper
MMU_S3.tbv[21]	PVBusMapper
MMU_S3.tbv[22]	PVBusMapper
MMU_S3.tbv[23]	PVBusMapper
MMU_S3.tbv[24]	PVBusMapper

InstanceName	ComponentName
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
MMU_S3.tbu[55]	PVBusMapper
MMU_S3.tbu[56]	PVBusMapper
MMU_S3.tbu[57]	PVBusMapper
MMU_S3.tbu[58]	PVBusMapper
MMU_S3.tbu[59]	PVBusMapper
MMU_S3.tbu[5]	PVBusMapper

InstanceName	ComponentName
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

This model has the following MTI trace components:

Table 3-1131: 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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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

MMU_S3 contains the following CADI targets:

- MMU_S3

Limitations

- The model supports architectural features and registers matching rOp0-00eac0 and r1p0-00eac0.
- PMU does not implement all architecturally-mandated events
- The PMU is intended for demonstration purposes only
- No power control
- 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.
- PVBus has no representation of the cache stash operations and so are not supported.

- The SYSCO interface is not implemented
- The low power interface is not implemented
- 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 then 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, error message is thrown which can be disabled by setting `all_error_messages_through_trace` to true.



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 then used for subsequent sufficiently similar transactions without needing the intervention of the SMMU model (and so won't be traced, etc.)

Notes

- 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 used will be listed in 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.
- A bad configuration renders the model inactive.
- Some configuration 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, 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.
- 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.
- 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 one.
- Debug and real accesses to the registers must be 32 or 64 bits.
- MSIs are issued on the `qtw_pvbus_m` port using attributes determined by the parameter `msi_attribute_transform`, whilst Event Queue writes will always be 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 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 will be forced to iNC-oNC-osh. For example, if a NoSnoop transaction uses a page table and is transformed to iWB-oWB-nsh then it will be 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 will exit 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 are interpreted as ! NoSnoop
 - Thus normalized to iWB-oWB-ish
 - Anything else is considered NoSnoop
 - Thus 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 will replicate the outer attribute into the inner attribute and then perform the normalisation 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. The normalisation is:

- If the transaction is classified as a PCIe one then the NoSnoop transform described above is applied. That is:
 - If the original transaction was NoSnoop then any weaker memory type is strengthened to iNC-oNC-osh, then 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.
- 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.
- 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.
- There is no functionality associated with TCU_NODE_STATUS register field ATSv, which indicates whether the node implements DTI-ATSv4.

Ports for MMU_S3

Table 3-1132: 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.

Name	Protocol	Type	Description
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 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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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. 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.



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`.



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_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]
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], ...
```

The `streamID` (32 b) is valid if `sidv` is 1 or both `sidv` and `nsidv` are unused.

`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].

The `substreamID` (20 b) is valid if `ssv` is true.

`SIDV == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

`nSEC_SID`, `nSSV`, `nSIDV` are available with negative logic. Different attributes are independent and can use negative or positive logic. 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`.

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]
```

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 `pvbush_id_routed_m` port as DTI-ATS is bidirectional, but PVBush 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

""

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/MasterID/UserFlags.

RHS Symbols: * MECID * 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. * 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"

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/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`.

msi_attribute_transform

Type

string

Default value

"ExtendedID[31:0]=smmu_msi_device_id, MasterID=0xFFFFFFFF"

Transform downstream attributes of MSI transactions.



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 was 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



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 the downstream attributes of a translated transaction.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV,
UserFlags[19:0]=SubstreamID,
MasterID[10]=MasterID[11], MasterID[11]=MasterID[10]"
```

RHS/LHS Symbols:

- ExtendedID/MasterID/UserFlags: incoming/outgoing attributes.

RHS Symbols:

- **DeviceID**
StreamID + translated_device_id_base
- StreamID/SubstreamID/SSV/SEC_SID
- **nSEC_SID/nSSV**
negative logic versions.
- **st1PBHA/st2PBHA**
Page Based Hardware Attributes from leaf descriptors (zero if unused).
- **STE_IMPDEF1**
STE[127:116]
- numeric literals.

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.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV, UserFlags[19:0]=SubstreamID,
MasterID[10]=MasterID[11], MasterID[11]=MasterID[10]"
```

RHS/LHS Symbols:

- **ExtendedID/MasterID/UserFlags**
incoming/outgoing attributes.

RHS Symbols:

- **DeviceID**
StreamID + translated_device_id_base
- StreamID/SubstreamID/SSV/SEC_SID
- **nSSV/nSEC_SID/**
negative logic versions.
- **St1PBHA/St2PBHA**
Page Based Hardware Attributes from leaf descriptors (zero if not used).
- **STE_IMPDEF1**
STE[127:116]
- **HWATTR_KIND_0**
PBHA information
- numeric literals.
- SIDV = 1, nSIDV = 0 (fixed values to indicate StreamID)

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_NoStreamID for NoStreamID transactions.

output_attribute_transform_for_NoStreamID

Type

string

Default value

"ExtendedID[31:0]=0, ExtendedID[32]=1"

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, MasterID[10]=MasterID[11],
MasterID[11]=MasterID[10] MasterID[9:6]=HWATTR_KIND_0"
```

RHS/LHS Symbols: * ExtendedID/MasterID/UserFlags: incoming/outgoing attributes.

RHS Symbols: * SIDV = 0, nSIDV = 1 (fixed values to indicate NoStreamID) * PAS *
HWATTR_KIND_0 * numeric literals.

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 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/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 use 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_l0gpt_entry_covers_log2size_in_bytes**Type**

uint32_t

Default value

30

Each LOGPT entry covers:

```
2**rme_l0gpt_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 `l0gptsz_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.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.

- "" or "none" – no transform

- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

- ExtendedID/MasterID/UserFlags

RHS Symbols:

- HWATTR_KIND_0: PBHA information
- 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
- numeric literals.

ExtendedID/MasterID/UserFlags start with values {0, 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 $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$.

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 $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$.

3.10.60 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 3-1134: 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 23.

Iris and MTI instances for MemoryMappedCounterModule

This model has the following Iris instances:

Table 3-1135: 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 3-1136: MemoryMappedCounterModule MTI instances

InstanceName	ComponentName
MemoryMappedCounterModule	MemoryMappedCounterModule
MemoryMappedCounterModule.pvbus_control_s[0]	PVBusSlave
MemoryMappedCounterModule.pvbus_read_s[0]	PVBusSlave

MemoryMappedCounterModule contains the following CADI targets:

- MemoryMappedCounterModule

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.



Note

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.

Ports for MemoryMappedCounterModule

Table 3-1137: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	This clock input determines the frequency of the Physical Count provided to the clusters connected to the cntvalueb port.
cntvalueb	CounterInterface	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	Signal	Slave	Resets when set.
pvbus_control_s	PVBus	Slave	This slave port provides memory-mapped read write access to the control registers of the module.
pvbus_read_s	PVBus	Slave	This slave port provides memory-mapped read access to the status frame registers.

Parameters for MemoryMappedCounterModule

base_frequency

Type

int

Default value

0x5f5e100

Reset value for CNTFID0, base frequency in Hz.

cntcldr0123_C

Type

int

Default value

0x0

Values to be returned for control-frame CIDR registers.

cntcldr0123_R

Type

int

Default value

0x0

Values to be returned for read-frame CIDR registers.

cntpidr0123_C

Type

int

Default value

0x0

Values to be returned for control-frame PIDR registers 0-3.

cntpidr0123_R

Type

int

Default value

0x0

Values to be returned for read-frame PIDR registers 0-3.

cntpidr4567_C**Type**

int

Default value

0x0

Values to be returned for control-frame PIDR registers 4-7.

cntpidr4567_R**Type**

int

Default value

0x0

Values to be returned for read-frame PIDR registers 4-7.

diagnostics**Type**

int

Default value

0x0

Diagnostics.

has_additional_registers**Type**

bool

Default value

0x0

Implements additional REFCLK CNT control registers.

has_counter_scaling**Type**

bool

Default value

0x0

Implements ARMv8.4 generic counter scaling (FEAT_CNTSC).

non_arch_fixed_frequency**Type**

int

Default value

0x0

If set, ignore CNTFID0 and instead use this frequency in Hz.

non_arch_start_at_default**Type**

bool

Default value

0x0

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

Ignore (rather than failing) on writes to read-frame.

use_real_time**Type**

bool

Default value

0x0

****Deprecated, this parameter will be removed in future versions**** Update the Generic Timer counter at a real-time base frequency instead of simulator time.

3.10.61 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 3-1138: 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 23.

Iris and MTI instances for MessageHandlingUnit

This model has the following Iris instances:

Table 3-1139: MessageHandlingUnit Iris instances

InstanceName	ComponentName
MessageHandlingUnit	MessageHandlingUnit
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

This model has the following MTI trace components:

Table 3-1140: 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

MessageHandlingUnit contains the following CADI targets:

- MessageHandlingUnit
- MessageHandlingUnitV2
- MessageHandlingUnitV3

Ports for MessageHandlingUnit

Table 3-1141: Ports

Name	Protocol	Type	Description
pvbus_s_rec	PVBus	Slave	-
pvbus_s_snd	PVBus	Slave	-
rec_combined_channel_irq_out[200]	Signal	Master	-
rec_combined_irq_out	Signal	Master	-
rec_reset_in	Signal	Slave	-

Name	Protocol	Type	Description
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

Number of doorbell channels.

NUM_FAST_CH

Type

int

Default value

0x1

Number of fast channels.

a_to_b_v3.NUM_FIFO_CH

Type

int

Default value

0x1

Number of FIFO Channels, default=1.

a_to_b_v3.auto_op_full

Type

bool

Default value

0x0

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

Fast Channel group interrupts enable, default=false.

a_to_b_v3.fifo_depth**Type**

int

Default value

0x4

Depth of the FIFO = fifo_depth + 1.

a_to_b_v3.m16ba_spt**Type**

bool

Default value

0x0

Mailbox 16 bit access support to FIFO registers.

a_to_b_v3.m32ba_spt**Type**

bool

Default value

0x1

Mailbox 32 bit access support to FIFO registers.

a_to_b_v3.m64ba_spt**Type**

bool

Default value

0x0

Mailbox 64 bit access support to FIFO registers.

a_to_b_v3.m8ba_spt**Type**

bool

Default value

0x0

Mailbox 8 bit access support to FIFO registers.

a_to_b_v3.monolithic**Type**

bool

Default value

0x1

Monolithic or Distributed MHU - default: monolithic(true).

a_to_b_v3.p16ba_spt**Type**

bool

Default value

0x0

Postbox 16 bit access support to FIFO registers.

a_to_b_v3.p32ba_spt**Type**

bool

Default value

0x1

Postbox 32 bit access support to FIFO registers.

a_to_b_v3.p64ba_spt**Type**

bool

Default value

0x0

Postbox 64 bit access support to FIFO registers.

a_to_b_v3.p8ba_spt**Type**

bool

Default value

0x0

Postbox 8 bit access support to FIFO registers.

diagnostics**Type**

int

Default value

0x2

Diagnostics 0==FATAL_ERROR -> 4==DEBUG.

fast_ch_group_int_enable**Type**

bool

Default value

0x0

Fast Channel group interrupts enable, default=false.

fast_ch_n_per_group**Type**

int

Default value

0x1

Fast Channel num channels per group, default=1.

fast_ch_num_groups**Type**

int

Default value

0x1

Fast Channel num of groups, default=1.

fast_ch_word_size**Type**

int

Default value

0x20

Fast Channel word size 32bit or 64bit, default=32.

major_version

Type
int

Default value
0x2

MHU major version (default=2).

mhu_arch_beta01

Type
bool

Default value
0x0

true = Aligns to MHUv3.beta01; false = Aligns to MHUv3.2.

minor_version

Type
int

Default value
0x1

MHU minor version (default=1).

product_id

Type
int

Default value
0x0

MHU part number.

3.10.62 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 3-1142: 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 23.

Iris and MTI instances for MessageHandlingUnitV2

This model has the following Iris instances:

Table 3-1143: MessageHandlingUnitV2 Iris instances

InstanceName	ComponentName
MessageHandlingUnitV2	MessageHandlingUnitV2

This model has the following MTI trace components:

Table 3-1144: MessageHandlingUnitV2 MTI instances

InstanceName	ComponentName
MessageHandlingUnitV2	MessageHandlingUnitV2

MessageHandlingUnitV2 contains the following CADI targets:

- MessageHandlingUnitV2

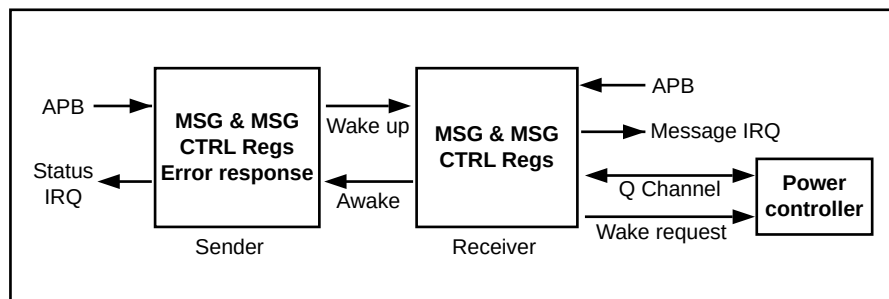
About MessageHandlingUnitV2

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



Ports for MessageHandlingUnitV2

Table 3-1145: Ports

Name	Protocol	Type	Description
int_access_nr2r	Signal	Master	-

Name	Protocol	Type	Description
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

Number of device channels.

minor_revision

Type

int

Default value

0x0

MHUv2 minor revision, 0 for v2.0, 1 for v2.1.

product_id

Type

int

Default value

0x0

MHUv2 Product ID, MHU Part Number.

3.10.63 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 3-1146: IP revisions support

Revision	Quality level
v3	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for MessageHandlingUnitV3

This model has the following Iris instances:

Table 3-1147: MessageHandlingUnitV3 Iris instances

InstanceName	ComponentName
MessageHandlingUnitV3	MessageHandlingUnitV3

This model has the following MTI trace components:

Table 3-1148: MessageHandlingUnitV3 MTI instances

InstanceName	ComponentName
MessageHandlingUnitV3	MessageHandlingUnitV3

MessageHandlingUnitV3 contains the following CADI targets:

- MessageHandlingUnitV3

Ports for MessageHandlingUnitV3

Table 3-1149: Ports

Name	Protocol	Type	Description
pdbus_s_rec	PVBus	Slave	Register access for Receiver/Mailbox
pdbus_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

Number of DoorBell Channels, default=1.

NUM_FAST_CH

Type

int

Default value

0x1

Number of Fast Channels, default=1.

NUM_FIFO_CH

Type

int

Default value

0x1

Number of FIFO Channels, default=1.

auto_op_full

Type

bool

Default value

0x0

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

diagnostics

Type

int

Default value

0x2

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

Fast Channel group interrupts enable, default=false.

fast_ch_n_per_group**Type**

int

Default value

0x1

Fast Channel num channels per group, default=1.

fast_ch_num_groups**Type**

int

Default value

0x1

Fast Channel num of groups, default=1.

fast_ch_word_size**Type**

int

Default value

0x20

Fast Channel word size 32bit or 64bit, default=32.

fifo_depth**Type**

int

Default value

0x4

Depth of the FIFO = fifo_depth + 1.

m16ba_spt**Type**

bool

Default value

0x0

Mailbox 16 bit access support to FIFO registers.

m32ba_spt**Type**

bool

Default value

0x1

Mailbox 32 bit access support to FIFO registers.

m64ba_spt**Type**

bool

Default value

0x0

Mailbox 64 bit access support to FIFO registers.

m8ba_spt**Type**

bool

Default value

0x0

Mailbox 8 bit access support to FIFO registers.

mhu_arch_beta01**Type**

bool

Default value

0x0

true = Aligns to MHUv3.beta01; false = Aligns to MHUv3.2.

monolithic**Type**

bool

Default value

0x1

Monolithic or Distributed MHU - default: monolithic(true).

p16ba_spt**Type**

bool

Default value

0x0

Postbox 16 bit access support to FIFO registers.

p32ba_spt**Type**

bool

Default value

0x1

Postbox 32 bit access support to FIFO registers.

p64ba_spt**Type**

bool

Default value

0x0

Postbox 64 bit access support to FIFO registers.

p8ba_spt**Type**

bool

Default value

0x0

Postbox 8 bit access support to FIFO registers.

3.10.64 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 3-1150: 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 23.

Iris and MTI instances for NI700

This model has the following Iris instances:

Table 3-1151: NI700 Iris instances

InstanceName	ComponentName
NI700	NI700
NI700.decoder	PVBusMapper

This model has the following MTI trace components:

Table 3-1152: NI700 MTI instances

InstanceName	ComponentName
NI700	NI700
NI700.decoder	PVBusMapper

NI700 contains the following CADI targets:

- NI700

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 ASNIIs `asni_s100_scp`, `asni_s101_dap`, and `asni_s204_periph0`:

- `asni_s100_scp` is mapped to `pvbus_s_asni[0]`
- `asni_s101_dap` is mapped to `pvbus_s_asni[1]`
- `asni_s204_periph0` is mapped to `pvbus_s_asni[2]`

Similarly:

- AMNIs are mapped to `pvbus_m_amni`
- HSNIs are mapped to `pvbus_s_hsn`
- HMNIs are mapped to `pvbus_m_hmni`
- PMNIs are mapped to `pvbus_m_pmni`

Additionally, NI700's parser prints the name-to-index mappings when the component parameter `print_parser_log=true`.

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

- 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.
- PMNI supports multiple interfaces but in the model only a single interface is listed in the `PMNI_INTERFACEID` registers of a PMNI.
- 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`.

Ports for NI700

Table 3-1153: 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.
pvbust_m_amni[127]	PVBus	Master	AMNI downstream ports.
pvbust_m_hmni[127]	PVBus	Master	HMNI downstream ports.
pvbust_m_pmni[127]	PVBus	Master	PMNI downstream ports.
pvbust_s_asni[128]	PVBus	Slave	ASNI upstream ports.
pvbust_s_hsni[128]	PVBus	Slave	HSNI upstream ports.
reset_in	Signal	Slave	Reset signal.

Parameters for NI700

mesh_config_file

Type

string

Default value

""

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

Value for PERIPHBASE.

print_config**Type**

bool

Default value

0x0

Enables printing the config register addresses.

print_parser_log**Type**

bool

Default value

0x0

Enables printing the yaml config parser log messages.

revision**Type**

string

Default value

"r2p0"

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

3.10.65 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 3-1154: IP revisions support

Revision	Quality level
r1p4	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for PL011_Uart

This model has the following Iris instances:

Table 3-1155: 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 3-1156: PL011_Uart MTI instances

InstanceName	ComponentName
PL011_Uart	PL011_Uart
PL011_Uart.busslave	PVBusSlave
PL011_Uart.clk_divider	ClockDivider

PL011_Uart contains the following CADI targets:

- ClockDivider
- ClockTimerThread
- ClockTimerThread64

- PL011_Uart
- SchedulerThread
- SchedulerThreadEvent

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
```



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.

Ports for PL011_Uart

Table 3-1157: Ports

Name	Protocol	Type	Description
<code>clk_in_ref</code>	<code>ClockSignal</code>	Slave	Clock input, typically 14.745MHz, which sets the master transmit/receive rate.
<code>intr</code>	<code>Signal</code>	Master	Interrupt signal.
<code>pvbuss</code>	<code>PVBus</code>	Slave	Slave port for register access.
<code>serial_out</code>	<code>SerialData</code>	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

Baud rate.

clk_divider.div

Type

int

Default value

0x1

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

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

clock_rate

Type

int

Default value

0xe10000

Clock rate for PL011.

enable_dc4

Type

bool

Default value

0x1

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

Enable hardware flow control workaround which forcefully disables CTSen and RTSen bits in UARTCR register.

halt**Type**

bool

Default value

0x0

Halt instead of shutdown for shutdown_on_eot and shutdown_tag.

in_file**Type**

string

Default value

""

Input file for data to be read by the UART.

in_file_escape_sequence**Type**

string

Default value

"##"

Input file escape sequence.

out_file**Type**

string

Default value

""

Output file to hold data written by the UART (use '-' to send all output to stdout).

revision**Type**

string

Default value

"r1p4"

Revision to simulate.

shutdown_on_eot**Type**

bool

Default value

0x0

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

""

Shutdown simulation when a string is transmitted.

toggle_mti**Type**

string

Default value

""

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

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

Unbuffered output.

untimed_fifos

Type

bool

Default value

0x1

Ignore the clock rate and transmit/receive serial data immediately.

3.10.66 PL011_Uart_cpp

ARM PrimeCell UART(PL011). This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1158: IP revisions support

Revision	Quality level
r1p4	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for PL011_Uart_cpp

This model has the following Iris instances:

Table 3-1159: PL011_Uart_cpp Iris instances

InstanceName	ComponentName
PL011_Uart_cpp	PL011_Uart
PL011_Uart_cpp.ClockDivider	ClockDivider
PL011_Uart_cpp.ClockTimerThread	ClockTimerThread
PL011_Uart_cpp.ClockTimerThread.ClockTimerThread64	ClockTimerThread64
PL011_Uart_cpp.ClockTimerThread.ClockTimerThread64.SchedulerThread	SchedulerThread
PL011_Uart_cpp.ClockTimerThread.ClockTimerThread64.SchedulerThreadEvent	SchedulerThreadEvent
PL011_Uart_cpp.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-1160: PL011_Uart_cpp MTI instances

InstanceName	ComponentName
PL011_Uart_cpp	PL011_Uart
PL011_Uart_cpp.ClockDivider	ClockDivider
PL011_Uart_cpp.PVBusSlave	PVBusSlave

PL011_Uart_cpp contains the following CADl targets:

- PL011_Uart

Ports for PL011_Uart_cpp

Table 3-1161: Ports

Name	Protocol	Type	Description
clk_in_ref	ClockSignal	Slave	-
intr	Signal	Master	-
pvbus	PVBus	Slave	-
serial_out	SerialData	Master	-

Parameters for PL011_Uart_cpp

baud_rate

Type

int

Default value

0x9600

Baud rate.

clock_rate

Type

int

Default value

0xe10000

Clock rate for PL011.

diagnostics

Type

int

Default value

0x0

Diagnostics.

enable_dc4**Type**

bool

Default value

0x1

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

Enable hardware flow control workaround which forcefully disables CTSen and RTSen bits in UARTCR register.

halt**Type**

bool

Default value

0x0

Halt instead of shutdown for shutdown_on_eot and shutdown_tag.

in_file**Type**

string

Default value

""

Input file for data to be read by the UART.

in_file_escape_sequence**Type**

string

Default value

"##"

Input file escape sequence.

out_file**Type**

string

Default value

""

Output file to hold data written by the UART (use '-' to send all output to stdout).

revision**Type**

string

Default value

"r1p4"

Revision to simulate.

shutdown_on_eot**Type**

bool

Default value

0x0

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

""

Shutdown simulation when a string is transmitted.

toggle_mti**Type**

string

Default value

""

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

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

Unbuffered output.

untimed_fifos

Type
bool

Default value
0x1

Ignore the clock rate and transmit/receive serial data immediately.

3.10.67 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 3-1162: IP revisions support

Revision	Quality level
r1p4	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for PL022_SSP

This model has the following Iris instances:

Table 3-1163: PL022_SSP Iris instances

InstanceName	ComponentName
PL022_SSP	PL022_SSP

InstanceName	ComponentName
PL022_SSP.busslave	PVBusSlave
PL022_SSP.prescaler	ClockDivider

This model has the following MTI trace components:

Table 3-1164: PL022_SSP MTI instances

InstanceName	ComponentName
PL022_SSP.busslave	PVBusSlave
PL022_SSP.prescaler	ClockDivider

PL022_SSP contains the following CADI targets:

- ClockDivider
- PL022_SSP

Differences between the model and the RTL



This component is a preliminary release. It is not a fully-supported peripheral.

Ports for PL022_SSP

Table 3-1165: 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

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

`prescaler.mul`

Type

int

Default value

0x1

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

3.10.68 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 3-1166: 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 23.

Iris and MTI instances for PL030_RTC

This model has the following Iris instances:

Table 3-1167: 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 3-1168: PL030_RTC MTI instances

InstanceName	ComponentName
PL030_RTC.busslave	PVBusSlave

PL030_RTC contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL030_RTC
- SchedulerThread
- SchedulerThreadEvent

Ports for PL030_RTC

Table 3-1169: Ports

Name	Protocol	Type	Description
clock	ClockSignal	Slave	Clock input, typically 1MHz, driving master count rate.
intr	Signal	Master	Interrupt signaling.
pvbuss	PVBus	Slave	Slave port for connection to PV bus master/decoder.

3.10.69 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 3-1170: IP revisions support

Revision	Quality level
r1p3	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for PL031_RTC

This model has the following Iris instances:

Table 3-1171: 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 3-1172: PL031_RTC MTI instances

InstanceName	ComponentName
PL031_RTC.busslave	PVBusSlave

PL031_RTC contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL031_RTC
- SchedulerThread
- SchedulerThreadEvent

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.

Ports for PL031_RTC

Table 3-1173: Ports

Name	Protocol	Type	Description
clock	ClockSignal	Slave	Clock input, typically 1MHz, driving master count rate.
intr	Signal	Master	Interrupt signaling.
pvbuss	PVBus	Slave	Slave port for connection to PV bus master/decoder.

Parameters for PL031_RTC

RTCDR_reset_value

Type

int

Default value

0x0

Reset value for RTCDR.

RTCDR_use_current_time

Type

bool

Default value

0x1

Use current Unix/POSIX time for reset value for RTCDR. If true RTCDR_reset_value is ignored.

3.10.70 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 3-1174: 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 23.

Iris and MTI instances for PL041_AACI

This model has the following Iris instances:

Table 3-1175: 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 3-1176: PL041_AACI MTI instances

InstanceName	ComponentName
PL041_AACI.busslave	PVBusSlave

PL041_AACI contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL041_AACI
- SchedulerThread
- SchedulerThreadEvent

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

Ports for PL041_AACI

Table 3-1177: 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

Host interface connection enabled.

3.10.71 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 3-1178: 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 23.

Iris and MTI instances for PL050_KMI

This model has the following Iris instances:

Table 3-1179: 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 3-1180: PL050_KMI MTI instances

InstanceName	ComponentName
PL050_KMI.busslave	PVBusSlave
PL050_KMI.clk_divider	ClockDivider

PL050_KMI contains the following CADI targets:

- ClockDivider
- PL050_KMI

About PL050_KMI

This model communicates with models of PS/2-like devices, for example a PS2Keyboard or PS2Mouse.

Ports for PL050_KMI

Table 3-1181: Ports

Name	Protocol	Type	Description
clock	ClockSignal	Slave	Clock input, typically 1MHz, which sets the master transmit/receive rate.
intr	Signal	Master	Master port signaling completion of transmit or receive.
ps2device	PS2Data	Slave	Used to communicate with a PS/2-like device.
pvbus	PVBus	Slave	Slave port for connection to PV bus master/decoder.

Parameters for PL050_KMI

`clk_divider.div`

Type

int

Default value

0x1

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

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

3.10.72 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 3-1182: 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 23.

Iris and MTI instances for PL061_GPIO

This model has the following Iris instances:

Table 3-1183: PL061_GPIO Iris instances

InstanceName	ComponentName
PL061_GPIO	PL061_GPIO
PL061_GPIO.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-1184: PL061_GPIO MTI instances

InstanceName	ComponentName
PL061_GPIO.busslave	PVBusSlave

PL061_GPIO contains the following CADI targets:

- PL061_GPIO

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.

Ports for PL061_GPIO

Table 3-1185: 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.
pvbus	PVBus	Slave	Slave port for register access.

Parameters for PL061_GPIO

init_inputs

Type

int

Default value

0x0

Default input values [7:0].

3.10.73 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 3-1186: IP revisions support

Revision	Quality level
r1p3	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for PL080_DMAC

This model has the following Iris instances:

Table 3-1187: 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 3-1188: PL080_DMAC MTI instances

InstanceName	ComponentName
PL080_DMAC.busmaster0	PVBusMaster
PL080_DMAC.busmaster1	PVBusMaster
PL080_DMAC.busslave	PVBusSlave

PL080_DMAC contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL080_DMAC
- SchedulerThread
- SchedulerThreadEvent

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.

Ports for PL080_DMAC

Table 3-1189: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock signal to control DMA transfer rate.

Name	Protocol	Type	Description
dma_port[16]	PL080_DMAC_DmaPortProtocol	Slave	request/response ports for communicating with devices.
interr	Signal	Master	DMA error interrupt signal.
intr	Signal	Master	Combined DMA error and terminal count signal.
inttc	Signal	Master	DMA terminal count signal.
pvbus0_m	PVBus	Master	Master bus interface 0 for DMA transfers.
pvbus1_m	PVBus	Master	Master bus interface 1 for DMA transfers.
pvbus_s	PVBus	Slave	Slave port for register accesses.
reset_in	Signal	Slave	System reset.

Parameters for PL080_DMAC

activate_delay

Type

int

Default value

0x0

request delay.

fifo_size

Type

int

Default value

0x10

Channel FIFO size in bytes.

generate_clear

Type

bool

Default value

0x0

Generate clear response.

max_transfer

Type

int

Default value

0x100

Largest atomic transfer.

3.10.74 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 3-1190: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for PL110_CLCD

This model has the following Iris instances:

Table 3-1191: 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
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 3-1192: PL110_CLCD MTI instances

InstanceName	ComponentName
PL110_CLCD.pl11x_clcd	PL11x_CLCD
PL110_CLCD.pl11x_clcd.busmaster	PVBusMaster
PL110_CLCD.pl11x_clcd.busslave	PVBusSlave

PL110_CLCD contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL110_CLCD
- PL11x_CLCD
- SchedulerThread

- SchedulerThreadEvent

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.

Ports for PL110_CLCD

Table 3-1193: 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.
pvbust	PVBUS	Slave	Slave port for register access.
pvbust_m	PVBUS	Master	DMA port for video data.

Parameters for PL110_CLCD

disable_snooping_dma

Type

bool

Default value

0x0

Disable DMA snooping.

pixel_double_limit

Type

int

Default value

0x12c

Minimum LCD pixel width before display will be zoomed.

3.10.75 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 3-1194: IP revisions support

Revision	Quality level
r0p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for PL111_CLCD

This model has the following Iris instances:

Table 3-1195: 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 3-1196: PL111_CLCD MTI instances

InstanceName	ComponentName
PL111_CLCD.pl11x_clcd	PL11x_CLCD
PL111_CLCD.pl11x_clcd.busmaster	PVBusMaster
PL111_CLCD.pl11x_clcd.busslave	PVBusSlave

PL111_CLCD contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL111_CLCD
- PL11x_CLCD
- SchedulerThread
- SchedulerThreadEvent

About PL111_CLCD

This component implements the hardware cursor support of the PL111_CLCD, which is the main difference with PL110_CLCD.

Ports for PL111_CLCD

Table 3-1197: 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.
pvbuss	PVBus	Slave	Slave port for register access.
pvbuss_m	PVBus	Master	DMA port for video data.

Parameters for PL111_CLCD

disable_snooping_dma

Type

bool

Default value

0x0

Disable DMA snooping.

pixel_double_limit

Type

int

Default value

0x12c

Minimum LCD pixel width before display will be zoomed.

3.10.76 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 3-1198: 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 23.

Iris and MTI instances for PL180_MCI

This model has the following Iris instances:

Table 3-1199: PL180_MCI Iris instances

InstanceName	ComponentName
PL180_MCI	PL180_MCI
PL180_MCI.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-1200: PL180_MCI MTI instances

InstanceName	ComponentName
PL180_MCI.busslave	PVBusSlave

PL180_MCI contains the following CADI targets:

- PL180_MCI

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.



Note

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.

Ports for PL180_MCI

Table 3-1201: 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

`p1180_fifo_depth`

Type
int

Default value
0x10

PL180 FIFO Depth.

3.10.77 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 3-1202: 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 23.

Iris and MTI instances for PL192_VIC

This model has the following Iris instances:

Table 3-1203: PL192_VIC Iris instances

InstanceName	ComponentName
PL192_VIC	PL192_VIC
PL192_VIC.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-1204: PL192_VIC MTI instances

InstanceName	ComponentName
PL192_VIC.busslave	PVBusSlave

PL192_VIC contains the following CADI targets:

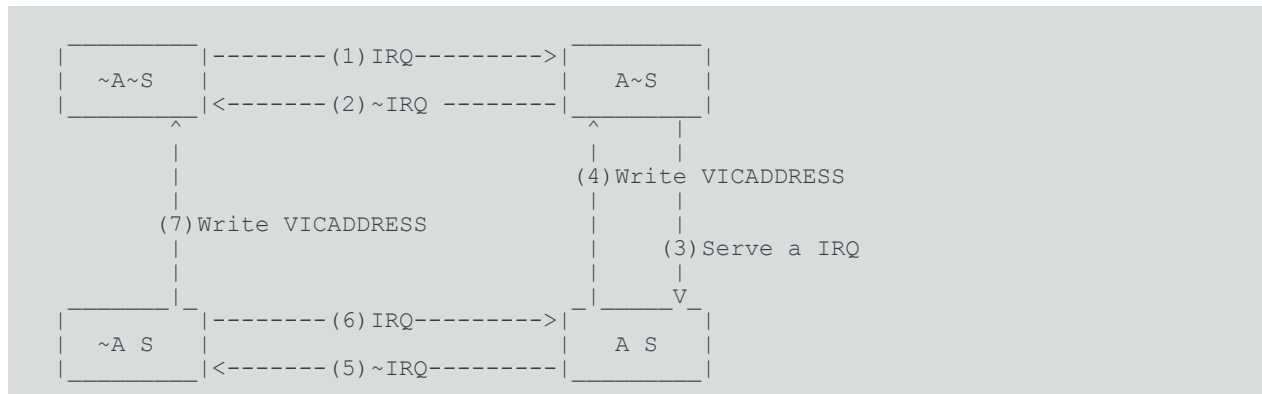
- PL192_VIC

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

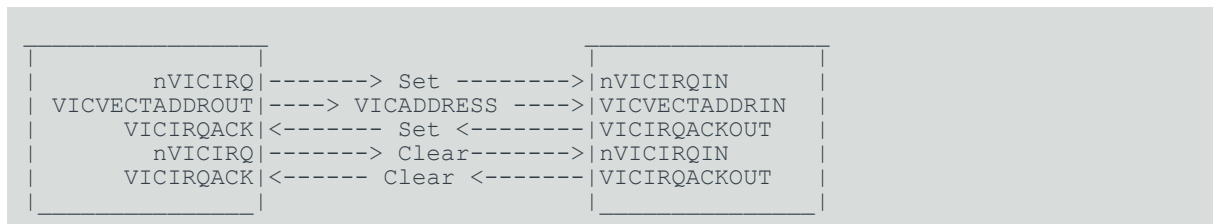
The IRQ is being served, and it is not in the stack. At this point, the device reasserts the IRQ and the state of IRQ changes from \sim AS to AS. In another words, the IRQ is reinserted into the stack.

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.

Ports for PL192_VIC

Table 3-1205: 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.
pvbuss	PVBus	Slave	Slave port for register access.

Name	Protocol	Type	Description
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.

3.10.78 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 3-1206: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for PL310_L2CC

This model has the following Iris instances:

Table 3-1207: PL310_L2CC Iris instances

InstanceName	ComponentName
PL310_L2CC	PL310_L2CC

This model has the following MTI trace components:

Table 3-1208: PL310_L2CC MTI instances

InstanceName	ComponentName
PL310_L2CC	PL310_L2CC

PL310_L2CC contains the following CADI targets:

- [PL310_L2CC](#)

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.



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.

Ports for PL310_L2CC

Table 3-1209: 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

Associativity for Auxiliary Control Register.

CACHEID**Type**

int

Default value

0x0

Cache controller cache ID.

CFGBIGEND**Type**

int

Default value

0x0

Big-endian mode for accessing configuration registers out of reset.

LOCKDOWN_BY_LINE**Type**

int

Default value

0x0

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

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

Base address for accessing configuration registers.

WAYSIZE**Type**

int

Default value

0x1

Size of ways for Auxiliary Control Register.

cache-state_modelled**Type**

bool

Default value

0x0

Specifies whether real cache state is modelled (vs. register model).

delay_cache_hit**Type**

int

Default value

0x0

Cost to handle a cache hit.

delay_cache_miss**Type**

int

Default value

0x0

Cost to handle a cache miss.

delay_cache_perbeat**Type**

int

Default value

0x0

Cost to handle one beat of cache data movement.

3.10.79 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 3-1210: IP revisions support

Revision	Quality level
rOp0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for PL330_DMAC

This model has the following Iris instances:

Table 3-1211: 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 3-1212: PL330_DMAC MTI instances

InstanceName	ComponentName
PL330_DMAC	PL330_DMAC
PL330_DMAC.busmaster	PVBusMaster
PL330_DMAC.busslave	PVBusSlave
PL330_DMAC.busslave_ns	PVBusSlave

PL330_DMAC contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL330_DMAC
- SchedulerThread
- SchedulerThreadEvent

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:

```
pl330_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 );
pl330_dma.pvbus_m => dma_labeller.pvbus_s;
dma_labeller.pvbus_m => output_bus.pvbus_s;
```

Ports for PL330_DMAC

Table 3-1213: 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.
pvbus_m	PVBus	Master	Master port for all memory accesses.
pvbus_s	PVBus	Slave	Slave port for all register accesses (secure).

Name	Protocol	Type	Description
pvbuss_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

request delay.

fifo_size

Type

int

Default value

0x10

Channel FIFO size in bytes.

generate_clear

Type

bool

Default value

0x0

Generate clear response.

max_transfer

Type

int

Default value

0x100

Largest atomic transfer.

p_axi_bus_width_param

Type

int

Default value

0x20

AXI bus width.

p_buffer_depth**Type**

int

Default value

0x10

buffer depth.

p_cache_line_words**Type**

int

Default value

0x1

number of words in a cache line.

p_cache_lines**Type**

int

Default value

0x1

number of cache lines.

p_controller_boots**Type**

bool

Default value

0x1

DMA boots from reset.

p_controller_nsecure**Type**

bool

Default value

0x0

Controller non-secure at reset (boot_manager_ns).

p_irq_nsecure**Type**

int

Default value

0x0

Interrupts non-secure at reset.

p_lsq_read_size**Type**

int

Default value

0x4

LSQ read buffer depth.

p_lsq_write_size**Type**

int

Default value

0x4

LSQ write buffer depth.

p_max_channels**Type**

int

Default value

0x8

virtual channels.

p_max_irqs**Type**

int

Default value

0x20

number of interrupts.

p_max_periph**Type**

int

Default value

0x20

number of peripheral interfaces.

p_perip_request_acceptance_0**Type**

int

Default value

0x2

Peripheral 0 request acceptance.

p_perip_request_acceptance_1**Type**

int

Default value

0x2

Peripheral 1 request acceptance.

p_perip_request_acceptance_10**Type**

int

Default value

0x2

Peripheral 10 request acceptance.

p_perip_request_acceptance_11**Type**

int

Default value

0x2

Peripheral 11 request acceptance.

p_perip_request_acceptance_12**Type**

int

Default value

0x2

Peripheral 12 request acceptance.

p_perip_request_acceptance_13**Type**

int

Default value

0x2

Peripheral 13 request acceptance.

p_perip_request_acceptance_14**Type**

int

Default value

0x2

Peripheral 14 request acceptance.

p_perip_request_acceptance_15**Type**

int

Default value

0x2

Peripheral 15 request acceptance.

p_perip_request_acceptance_16**Type**

int

Default value

0x2

Peripheral 16 request acceptance.

p_perip_request_acceptance_17**Type**

int

Default value

0x2

Peripheral 17 request acceptance.

p_perip_request_acceptance_18**Type**

int

Default value

0x2

Peripheral 18 request acceptance.

p_perip_request_acceptance_19**Type**

int

Default value

0x2

Peripheral 19 request acceptance.

p_perip_request_acceptance_2**Type**

int

Default value

0x2

Peripheral 2 request acceptance.

p_perip_request_acceptance_20**Type**

int

Default value

0x2

Peripheral 20 request acceptance.

p_perip_request_acceptance_21**Type**

int

Default value

0x2

Peripheral 21 request acceptance.

p_perip_request_acceptance_22**Type**

int

Default value

0x2

Peripheral 22 request acceptance.

p_perip_request_acceptance_23**Type**

int

Default value

0x2

Peripheral 23 request acceptance.

p_perip_request_acceptance_24**Type**

int

Default value

0x2

Peripheral 24 request acceptance.

p_perip_request_acceptance_25**Type**

int

Default value

0x2

Peripheral 25 request acceptance.

p_perip_request_acceptance_26**Type**

int

Default value

0x2

Peripheral 26 request acceptance.

p_perip_request_acceptance_27**Type**

int

Default value

0x2

Peripheral 27 request acceptance.

p_perip_request_acceptance_28**Type**

int

Default value

0x2

Peripheral 28 request acceptance.

p_perip_request_acceptance_29**Type**

int

Default value

0x2

Peripheral 29 request acceptance.

p_perip_request_acceptance_3**Type**

int

Default value

0x2

Peripheral 3 request acceptance.

p_perip_request_acceptance_30**Type**

int

Default value

0x2

Peripheral 30 request acceptance.

p_perip_request_acceptance_31**Type**

int

Default value

0x2

Peripheral 31 request acceptance.

p_perip_request_acceptance_4**Type**

int

Default value

0x2

Peripheral 4 request acceptance.

p_perip_request_acceptance_5**Type**

int

Default value

0x2

Peripheral 5 request acceptance.

p_perip_request_acceptance_6**Type**

int

Default value

0x2

Peripheral 6 request acceptance.

p_perip_request_acceptance_7**Type**

int

Default value

0x2

Peripheral 7 request acceptance.

p_perip_request_acceptance_8**Type**

int

Default value

0x2

Peripheral 8 request acceptance.

p_perip_request_acceptance_9**Type**

int

Default value

0x2

Peripheral 9 request acceptance.

p_periph_nsecure**Type**

bool

Default value

0x0

Peripherals non-secure at reset.

p_read_issuing_capability**Type**

int

Default value

0x1

AXI read issuing capability.

p_reset_pc

Type
int

Default value
0x60000000

DMA PC at reset.

p_write_issuing_capability

Type
int

Default value
0x1

AXI write issuing capability.

revision

Type
string

Default value
"r0p0"

revision ID.

3.10.80 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 3-1214: 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 23.

Iris and MTI instances for PL340_DMC

This model has the following Iris instances:

Table 3-1215: PL340_DMC Iris instances

InstanceName	ComponentName
PL340_DMC	PL340_DMC
PL340_DMC.apb_slave	PVBusSlave

InstanceName	ComponentName
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

This model has the following MTI trace components:

Table 3-1216: 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

PL340_DMC contains the following CADI targets:

- PL340_DMC
- PVBusExclusiveMonitor

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.

Ports for PL340_DMC

Table 3-1217: 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

Set this parameter to 0 if memory is connected.

IF_CHIP1

Type

int

Default value

0xffffffffffffffff

Set this parameter to 0 if memory is connected.

IF_CHIP2

Type

int

Default value

0xffffffffffffffff

Set this parameter to 0 if memory is connected.

IF_CHIP3

Type

int

Default value

0xffffffffffffffff

Set this parameter to 0 if memory is connected.

MEMORY_WIDTH

Type

int

Default value

0x20

Set this parameter to 0 if memory is connected.

exclusive_monitor0.apply_access_width_criteria_to_non_excl_stores**Type**

bool

Default value

0x1

Apply the given exclusive store width matching criteria to non-exclusive stores.

exclusive_monitor0.clear_on_strex_address_mismatch**Type**

bool

Default value

0x1

Whether monitor is cleared when strex fails due to address mismatch.

exclusive_monitor0.enable_component**Type**

bool

Default value

0x1

Enable component.

exclusive_monitor0.exclusive_monitor_clear_on_atomic_from_same_master**Type**

bool

Default value

0x1

Monitor atomics from the same master.

exclusive_monitor0.match_secure_state**Type**

bool

Default value

0x1

Treat the secure state like an address bit.

exclusive_monitor0.monitor_access_level**Type**

int

Default value

0x0

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

Monitor non-exclusive stores from the same master.

exclusive_monitor0.number_of_monitors**Type**

int

Default value

0x8

Number of monitors.

exclusive_monitor0.shareability_domain**Type**

int

Default value

0x3

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

Apply the given exclusive store width matching criteria to non-exclusive stores.

exclusive_monitor1.clear_on_strex_address_mismatch**Type**

bool

Default value

0x1

Whether monitor is cleared when strex fails due to address mismatch.

exclusive_monitor1.enable_component**Type**

bool

Default value

0x1

Enable component.

exclusive_monitor1.exclusive_monitor_clear_on_atomic_from_same_master**Type**

bool

Default value

0x1

Monitor atomics from the same master.

exclusive_monitor1.match_secure_state**Type**

bool

Default value

0x1

Treat the secure state like an address bit.

exclusive_monitor1.monitor_access_level**Type**

int

Default value

0x0

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

Monitor non-exclusive stores from the same master.

exclusive_monitor1.number_of_monitors**Type**

int

Default value

0x8

Number of monitors.

exclusive_monitor1.shareability_domain**Type**

int

Default value

0x3

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

Apply the given exclusive store width matching criteria to non-exclusive stores.

exclusive_monitor2.clear_on_strex_address_mismatch**Type**

bool

Default value

0x1

Whether monitor is cleared when strex fails due to address mismatch.

exclusive_monitor2.enable_component**Type**

bool

Default value

0x1

Enable component.

exclusive_monitor2.exclusive_monitor_clear_on_atomic_from_same_master**Type**

bool

Default value

0x1

Monitor atomics from the same master.

exclusive_monitor2.match_secure_state**Type**

bool

Default value

0x1

Treat the secure state like an address bit.

exclusive_monitor2.monitor_access_level**Type**

int

Default value

0x0

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

Monitor non-exclusive stores from the same master.

exclusive_monitor2.number_of_monitors**Type**

int

Default value

0x8

Number of monitors.

exclusive_monitor2.shareability_domain**Type**

int

Default value

0x3

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

Apply the given exclusive store width matching criteria to non-exclusive stores.

exclusive_monitor3.clear_on_strex_address_mismatch**Type**

bool

Default value

0x1

Whether monitor is cleared when strex fails due to address mismatch.

exclusive_monitor3.enable_component**Type**

bool

Default value

0x1

Enable component.

exclusive_monitor3.exclusive_monitor_clear_on_atomic_from_same_master**Type**

bool

Default value

0x1

Monitor atomics from the same master.

exclusive_monitor3.match_secure_state**Type**

bool

Default value

0x1

Treat the secure state like an address bit.

exclusive_monitor3.monitor_access_level**Type**

int

Default value

0x0

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

Monitor non-exclusive stores from the same master.

exclusive_monitor3.number_of_monitors**Type**

int

Default value

0x8

Number of monitors.

exclusive_monitor3.shareability_domain**Type**

int

Default value

0x3

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

3.10.81 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 3-1218: 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 23.

Iris and MTI instances for PL350_SMC

This model has the following Iris instances:

Table 3-1219: 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

InstanceName	ComponentName
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 3-1220: 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

InstanceName	ComponentName
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

PL350_SMC contains the following CADI targets:

- PL350_SMC
- PVBusExclusiveMonitor
- TZSwitch

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.

Ports for PL350_SMC

Table 3-1221: 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.
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

Interface 0 chip 0 connected.

IF0_CHIP0_BASE

Type

int

Default value

0x0

Interface 0 chip 0 Base address.

IF0_CHIP0_SIZE

Type

int

Default value

0x0

Interface 0 chip 0 Size.

IF0_CHIP1

Type

bool

Default value

0x0

Interface 0 chip 1 connected.

IF0_CHIP1_BASE

Type

int

Default value

0x0

Interface 0 chip 1 Base address.

IF0_CHIP1_SIZE**Type**

int

Default value

0x0

Interface 0 chip 1 Size.

IF0_CHIP2**Type**

bool

Default value

0x0

Interface 0 chip 2 connected.

IF0_CHIP2_BASE**Type**

int

Default value

0x0

Interface 0 chip 2 Base address.

IF0_CHIP2_SIZE**Type**

int

Default value

0x0

Interface 0 chip 2 Size.

IF0_CHIP3**Type**

bool

Default value

0x0

Interface 0 chip 3 connected.

IF0_CHIP3_BASE**Type**

int

Default value

0x0

Interface 0 chip 3 Base address.

IF0_CHIP3_SIZE**Type**

int

Default value

0x0

Interface 0 chip 3 Size.

IF0_MEM_TYPE_PARAMETER**Type**

int

Default value

0x0

Interface 0 Mem type.

IF1_CHIP0**Type**

bool

Default value

0x0

Interface 1 chip 0 connected.

IF1_CHIP0_BASE**Type**

int

Default value

0x0

Interface 1 chip 0 Base address.

IF1_CHIP0_SIZE**Type**

int

Default value

0x0

Interface 1 chip 0 Size.

IF1_CHIP1**Type**

bool

Default value

0x0

Interface 1 chip 1 connected.

IF1_CHIP1_BASE**Type**

int

Default value

0x0

Interface 1 chip 1 Base address.

IF1_CHIP1_SIZE**Type**

int

Default value

0x0

Interface 1 chip 1 Size.

IF1_CHIP2**Type**

bool

Default value

0x0

Interface 1 chip 2 connected.

IF1_CHIP2_BASE**Type**

int

Default value

0x0

Interface 1 chip 2 Base address.

IF1_CHIP2_SIZE**Type**

int

Default value

0x0

Interface 1 chip 2 Size.

IF1_CHIP3**Type**

bool

Default value

0x0

Interface 1 chip 3 connected.

IF1_CHIP3_BASE**Type**

int

Default value

0x0

Interface 1 chip 3 Base address.

IF1_CHIP3_SIZE**Type**

int

Default value

0x0

Interface 1 chip 3 Size.

IF1_MEM_TYPE_PARAMETER**Type**

int

Default value

0x0

Interface 1 Mem type.

PERIPH_ID_0**Type**

int

Default value

0x52

Periph_ID_0 value.

REMAP**Type**

int

Default value

0xffffffffffffffff

Remap the device.

addr_remapper.normal**Type**

int

Default value

0x2

Normal Port.

addr_remapper.secure**Type**

int

Default value

0x1

Secure Port.

exclusive_monitor0_0.apply_access_width_criteria_to_non_excl_stores**Type**

bool

Default value

0x1

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

Whether monitor is cleared when strex fails due to address mismatch.

exclusive_monitor0_0.enable_component**Type**

bool

Default value

0x1

Enable component.

exclusive_monitor0_0.exclusive_monitor_clear_on_atomic_from_same_master**Type**

bool

Default value

0x1

Monitor atomics from the same master.

exclusive_monitor0_0.match_secure_state**Type**

bool

Default value

0x1

Treat the secure state like an address bit.

exclusive_monitor0_0.monitor_access_level**Type**

int

Default value

0x0

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

Monitor non-exclusive stores from the same master.

exclusive_monitor0_0.number_of_monitors**Type**

int

Default value

0x8

Number of monitors.

exclusive_monitor0_0.shareability_domain**Type**

int

Default value

0x3

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

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

Whether monitor is cleared when strex fails due to address mismatch.

exclusive_monitor0_1.enable_component**Type**

bool

Default value

0x1

Enable component.

exclusive_monitor0_1.exclusive_monitor_clear_on_atomic_from_same_master**Type**

bool

Default value

0x1

Monitor atomics from the same master.

exclusive_monitor0_1.match_secure_state**Type**

bool

Default value

0x1

Treat the secure state like an address bit.

exclusive_monitor0_1.monitor_access_level**Type**

int

Default value

0x0

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

Monitor non-exclusive stores from the same master.

exclusive_monitor0_1.number_of_monitors**Type**

int

Default value

0x8

Number of monitors.

exclusive_monitor0_1.shareability_domain**Type**

int

Default value

0x3

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

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

Whether monitor is cleared when strex fails due to address mismatch.

exclusive_monitor0_2.enable_component**Type**

bool

Default value

0x1

Enable component.

exclusive_monitor0_2.exclusive_monitor_clear_on_atomic_from_same_master**Type**

bool

Default value

0x1

Monitor atomics from the same master.

exclusive_monitor0_2.match_secure_state**Type**

bool

Default value

0x1

Treat the secure state like an address bit.

exclusive_monitor0_2.monitor_access_level**Type**

int

Default value

0x0

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

Monitor non-exclusive stores from the same master.

exclusive_monitor0_2.number_of_monitors**Type**

int

Default value

0x8

Number of monitors.

exclusive_monitor0_2.shareability_domain**Type**

int

Default value

0x3

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

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

Whether monitor is cleared when strex fails due to address mismatch.

exclusive_monitor0_3.enable_component**Type**

bool

Default value

0x1

Enable component.

exclusive_monitor0_3.exclusive_monitor_clear_on_atomic_from_same_master**Type**

bool

Default value

0x1

Monitor atomics from the same master.

exclusive_monitor0_3.match_secure_state**Type**

bool

Default value

0x1

Treat the secure state like an address bit.

exclusive_monitor0_3.monitor_access_level**Type**

int

Default value

0x0

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

Monitor non-exclusive stores from the same master.

exclusive_monitor0_3.number_of_monitors**Type**

int

Default value

0x8

Number of monitors.

exclusive_monitor0_3.shareability_domain**Type**

int

Default value

0x3

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

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

Whether monitor is cleared when strex fails due to address mismatch.

exclusive_monitor1_0.enable_component**Type**

bool

Default value

0x1

Enable component.

exclusive_monitor1_0.exclusive_monitor_clear_on_atomic_from_same_master**Type**

bool

Default value

0x1

Monitor atomics from the same master.

exclusive_monitor1_0.match_secure_state**Type**

bool

Default value

0x1

Treat the secure state like an address bit.

exclusive_monitor1_0.monitor_access_level**Type**

int

Default value

0x0

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

Monitor non-exclusive stores from the same master.

exclusive_monitor1_0.number_of_monitors**Type**

int

Default value

0x8

Number of monitors.

exclusive_monitor1_0.shareability_domain**Type**

int

Default value

0x3

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

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

Whether monitor is cleared when strex fails due to address mismatch.

exclusive_monitor1_1.enable_component**Type**

bool

Default value

0x1

Enable component.

exclusive_monitor1_1.exclusive_monitor_clear_on_atomic_from_same_master**Type**

bool

Default value

0x1

Monitor atomics from the same master.

exclusive_monitor1_1.match_secure_state**Type**

bool

Default value

0x1

Treat the secure state like an address bit.

exclusive_monitor1_1.monitor_access_level**Type**

int

Default value

0x0

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

Monitor non-exclusive stores from the same master.

exclusive_monitor1_1.number_of_monitors**Type**

int

Default value

0x8

Number of monitors.

exclusive_monitor1_1.shareability_domain**Type**

int

Default value

0x3

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

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

Whether monitor is cleared when strex fails due to address mismatch.

exclusive_monitor1_2.enable_component**Type**

bool

Default value

0x1

Enable component.

exclusive_monitor1_2.exclusive_monitor_clear_on_atomic_from_same_master**Type**

bool

Default value

0x1

Monitor atomics from the same master.

exclusive_monitor1_2.match_secure_state**Type**

bool

Default value

0x1

Treat the secure state like an address bit.

exclusive_monitor1_2.monitor_access_level**Type**

int

Default value

0x0

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

Monitor non-exclusive stores from the same master.

exclusive_monitor1_2.number_of_monitors**Type**

int

Default value

0x8

Number of monitors.

exclusive_monitor1_2.shareability_domain**Type**

int

Default value

0x3

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

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

Whether monitor is cleared when strex fails due to address mismatch.

exclusive_monitor1_3.enable_component**Type**

bool

Default value

0x1

Enable component.

exclusive_monitor1_3.exclusive_monitor_clear_on_atomic_from_same_master**Type**

bool

Default value

0x1

Monitor atomics from the same master.

exclusive_monitor1_3.match_secure_state**Type**

bool

Default value

0x1

Treat the secure state like an address bit.

exclusive_monitor1_3.monitor_access_level**Type**

int

Default value

0x0

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

Monitor non-exclusive stores from the same master.

exclusive_monitor1_3.number_of_monitors

Type
int

Default value
0x8

Number of monitors.

exclusive_monitor1_3.shareability_domain

Type
int

Default value
0x3

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"

Revision.

3.10.82 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 3-1222: 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 23.

Iris and MTI instances for PL350_SMC_NAND_FLASH

This model has the following Iris instances:

Table 3-1223: 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 3-1224: PL350_SMC_NAND_FLASH MTI instances

InstanceName	ComponentName
PL350_SMC_NAND_FLASH.busslave	PVBusSlave

PL350_SMC_NAND_FLASH contains the following CADI targets:

- PL350_SMC_NAND_FLASH

About PL350_SMC_NAND_FLASH

Program the component as you would the hardware.

Ports for PL350_SMC_NAND_FLASH

Table 3-1225: Ports

Name	Protocol	Type	Description
irq	Signal	Master	Interrupt signaling.
pvbuss	PVBus	Slave	Slave port for connection to PV bus master/decoder.

Parameters for PL350_SMC_NAND_FLASH

DEVICE_1

Type

int

Default value

0xec

Device manufacturer code.

DEVICE_2

Type

int

Default value

0xda

Device code.

DEVICE_3**Type**

int

Default value

0x80

Device 3rd cycle code.

DEVICE_4**Type**

int

Default value

0x15

Device 4th cycle code.

DEVICE_NAME**Type**

string

Default value

"Samsung K9F1G08U0M"

Device Name.

NAND_BLOCK_COUNT**Type**

int

Default value

0x800

number of blocks in the flash device.

NAND_FLASH_SIZE**Type**

int

Default value

0x10800000

flash size in byte.

NAND_PAGE_COUNT_PER_BLOCK

Type
int

Default value
0x40

number of pages in each block.

NAND_PAGE_SIZE

Type
int

Default value
0x840

page size.

NAND_SPARE_SIZE_PER_PAGE

Type
int

Default value
0x40

Spare size per page.

NAND_VALID_SIZE_PER_PAGE

Type
int

Default value
0x800

valid page size.

3.10.83 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 3-1226: 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 23.

Iris and MTI instances for PL370_HDLCD

This model has the following Iris instances:

Table 3-1227: 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 3-1228: PL370_HDLCD MTI instances

InstanceName	ComponentName
PL370_HDLCD.busmaster	PVBusMaster
PL370_HDLCD.busslave	PVBusSlave

PL370_HDLCD contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL370_HDLCD
- SchedulerThread
- SchedulerThreadEvent



Too fast a pixel clock can slow the rest of the simulation.

Ports for PL370_HDLCD

Table 3-1229: 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.
intr	Signal	Master	Interrupt signaling line for flyback events.
pvbus	PVBus	Slave	Slave port for connection to PV bus master/decoder.

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	DMA port for collecting video data from memory/framebuffer.

Parameters for PL370_HDLCD

diagnostics

Type

int

Default value

0x0

Diagnostics level.

disable_snooping_dma

Type

bool

Default value

0x0

Disable DMA snooping.

force_frame_rate

Type

int

Default value

0x32

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.

3.10.84 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 3-1230: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for PL390_GIC

This model has the following Iris instances:

Table 3-1231: 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 3-1232: PL390_GIC MTI instances

InstanceName	ComponentName
PL390_GIC.busslave_cpu	PVBusSlave
PL390_GIC.busslave_distributor	PVBusSlave

PL390_GIC contains the following CADI targets:

- PL390_GIC

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 `pdbus_cpu` and `pdbus_distributor`. The `enable_cx` and `match_cx` signals identify the originator of a transaction on `pdbus_cpu`. Similarly, the `enable_dx` and `match_dx` signals identify the originator of a transaction on `pdbus_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
```


Ports for PL390_GIC

Table 3-1233: Ports

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	Set preventing write accesses to security-critical configuration registers.
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

set architecture version in periph_id register.

AXI_IF

Type

bool

Default value

0x1

set interface type in peripheral identification register 8.

C_ID_WIDTH**Type**

int

Default value

0x20

width of the cpu interface master id.

D_ID_WIDTH**Type**

int

Default value

0x20

width of the distributor interface master id.

ENABLE_LEGACY_FIQ**Type**

bool

Default value

0x1

provide legacy fiq interrupt inputs.

ENABLE_LEGACY_IRQ**Type**

bool

Default value

0x1

provide legacy irq interrupt inputs.

ENABLE_PPI_EDGE**Type**

bool

Default value

0x0

ppi edge sensitive.

ENABLE_TRUSTZONE**Type**

bool

Default value

0x1

support trustzone.

INIT_ENABLE_C0**Type**

int

Default value

0xffffffff

initial value of register ENABLE_C0.

INIT_ENABLE_C1**Type**

int

Default value

0xffffffff

initial value of register ENABLE_C1.

INIT_ENABLE_C2**Type**

int

Default value

0xffffffff

initial value of register ENABLE_C2.

INIT_ENABLE_C3**Type**

int

Default value

0xffffffff

initial value of register ENABLE_C3.

INIT_ENABLE_C4**Type**

int

Default value

0xffffffff

initial value of register ENABLE_C4.

INIT_ENABLE_C5**Type**

int

Default value

0xffffffff

initial value of register ENABLE_C5.

INIT_ENABLE_C6**Type**

int

Default value

0xffffffff

initial value of register ENABLE_C6.

INIT_ENABLE_C7**Type**

int

Default value

0xffffffff

initial value of register ENABLE_C7.

INIT_ENABLE_D0**Type**

int

Default value

0xffffffff

initial value of register ENABLE_D0.

INIT_ENABLE_D1**Type**

int

Default value

0xffffffff

initial value of register ENABLE_D1.

INIT_ENABLE_D2**Type**

int

Default value

0xffffffff

initial value of register ENABLE_D2.

INIT_ENABLE_D3**Type**

int

Default value

0xffffffff

initial value of register ENABLE_D3.

INIT_ENABLE_D4**Type**

int

Default value

0xffffffff

initial value of register ENABLE_D4.

INIT_ENABLE_D5**Type**

int

Default value

0xffffffff

initial value of register ENABLE_D5.

INIT_ENABLE_D6**Type**

int

Default value

0xffffffff

initial value of register ENABLE_D6.

INIT_ENABLE_D7**Type**

int

Default value

0xffffffff

initial value of register ENABLE_D7.

INIT_MATCH_C0**Type**

int

Default value

0x0

initial value of register MATCH_C0.

INIT_MATCH_C1**Type**

int

Default value

0x1

initial value of register MATCH_C1.

INIT_MATCH_C2**Type**

int

Default value

0x2

initial value of register MATCH_C2.

INIT_MATCH_C3**Type**

int

Default value

0x3

initial value of register MATCH_C3.

INIT_MATCH_C4**Type**

int

Default value

0x4

initial value of register MATCH_C4.

INIT_MATCH_C5**Type**

int

Default value

0x5

initial value of register MATCH_C5.

INIT_MATCH_C6**Type**

int

Default value

0x6

initial value of register MATCH_C6.

INIT_MATCH_C7**Type**

int

Default value

0x7

initial value of register MATCH_C7.

INIT_MATCH_D0**Type**

int

Default value

0x0

initial value of register MATCH_D0.

INIT_MATCH_D1**Type**

int

Default value

0x1

initial value of register MATCH_D1.

INIT_MATCH_D2**Type**

int

Default value

0x2

initial value of register MATCH_D2.

INIT_MATCH_D3**Type**

int

Default value

0x3

initial value of register MATCH_D3.

INIT_MATCH_D4**Type**

int

Default value

0x4

initial value of register MATCH_D4.

INIT_MATCH_D5**Type**

int

Default value

0x5

initial value of register MATCH_D5.

INIT_MATCH_D6**Type**

int

Default value

0x6

initial value of register MATCH_D6.

INIT_MATCH_D7**Type**

int

Default value

0x7

initial value of register MATCH_D7.

NUM_CPU**Type**

int

Default value

0x8

number of cpu interfaces.

NUM_LSPI**Type**

int

Default value

0x1f

number of lockable shared peripheral interrupts.

NUM_PPI

Type
int

Default value
0x10

number of peripheral interrupts.

NUM_PRIORITY_LEVELS

Type
int

Default value
0x100

number of priority levels.

NUM_SGI

Type
int

Default value
0x10

number of software generated interrupts.

NUM_SPI

Type
int

Default value
0x3dc

number of shared peripheral interrupts.

3.10.85 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 3-1234: 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 23.

Ports for PPUMTWakerequest

Table 3-1235: 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

Multi-threaded mode.

thread0_op_mode_bit

Type

int

Default value

0x10

Thread0 Operation Mode bit of DEVPACTIVE.

thread1_op_mode_bit

Type

int

Default value

0x11

Thread1 Operation Mode bit of DEVPACTIVE.

3.10.86 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 3-1236: 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 23.

Iris and MTI instances for PPUv0

This model has the following Iris instances:

Table 3-1237: PPUv0 Iris instances

InstanceName	ComponentName
PPUv0	PPUv0
PPUv0.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-1238: PPUv0 MTI instances

InstanceName	ComponentName
PPUv0	PPUv0
PPUv0.busslave	PVBusSlave

PPUv0 contains the following CADI targets:

- [PPUv0](#)

Ports for PPUv0

Table 3-1239: 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

Default power state ON.

device_channels

Type

int

Default value

0x0

Number of device channels (0: P-Channel, 1-8: Q-Channels).

dynamic_off

Type

bool

Default value

0x0

Dynamic Off.

dynamic_on

Type

bool

Default value

0x0

Dynamic On.

dynamic_warm_reset

Type

bool

Default value

0x0

Dynamic Warm Reset.

full_ret

Type

int

Default value

0x0

Full Retention (0: not supported, 1: static, 2: dynamic).

func_ret**Type**

int

Default value

0x0

Functional Retention (0: not supported, 1: static, 2: dynamic).

logic_ret**Type**

int

Default value

0x0

Logic Retention (0: not supported, 1: static, 2: dynamic).

mem_off**Type**

int

Default value

0x0

Memory Off (0: not supported, 1: static, 2: dynamic).

mem_ret**Type**

int

Default value

0x0

Memory Retention (0: not supported, 1: static, 2: dynamic).

revision**Type**

string

Default value

"rOp0"

Revision.

use_active_signal

Type
bool

Default value
0x0

Use device-active signal.

3.10.87 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 3-1240: 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 23.

Changes in 11.27.17

Parameters added:

- bypass_handshake

Iris and MTI instances for PPUv1

This model has the following Iris instances:

Table 3-1241: PPUv1 Iris instances

InstanceName	ComponentName
PPUv1	PPUv1
PPUv1.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-1242: PPUv1 MTI instances

InstanceName	ComponentName
PPUv1	PPUv1
PPUv1.busslave	PVBusSlave

PPUv1 contains the following CADI targets:

- PPUv1

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 `DEVPSTATE` 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 [2.5.1 PChannel protocol](#) on page 90 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 `PREQUEST` 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.

Ports for PPUv1

Table 3-1243: Ports

Name	Protocol	Type	Description
<code>dev_clk_en_out</code>	Signal	Master	Domain clock enable
<code>dev_emu_clk_en_out</code>	Signal	Master	Domain emulated mode clock enable
<code>dev_emu_isolaten_out</code>	Signal	Master	Domain emulated isolation control.
<code>dev_isolaten_out</code>	Signal	Master	Domain isolation control.
<code>dev_poresetn_out</code>	Signal	Master	Domain power on reset
<code>dev_ret_resetrn_out</code>	Signal	Master	Domain retention reset.
<code>dev_warm_resetrn_out</code>	Signal	Master	Domain warm reset
<code>devpactive</code>	PChannel	Master	P-Channel port
<code>irq</code>	Signal	Master	PPU IRQ signal
<code>powerdown</code>	Signal	Master	Notify whether or not the PPU is in OFF state.
<code>ppuhwstat</code>	Value	Master	Notify the power state change inside the PPU. The definition of each bit is the same as <code>DEVPACTIVE[10:0]</code> .
<code>pvbus_s</code>	PVBus	Slave	PPU APB bus slave port
<code>reset_in</code>	Signal	Slave	PPU reset signal input
<code>wakerequest</code>	Signal	Slave	Input port for the wakerequest signal. It is ORed with <code>PACTIVE[8]</code> (ON) inside the PPU as input to PPU <code>DEVPACTIVE[8]</code> (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

Whether to support Rev D locked IRQ.

bypass_handshake

Type

bool

Default value

0x0

Bypass pcsn handshake.

dbg_recov

Type

int

Default value

0x0

Debug Recovery Reset (0: not supported, 1: static mode only, 2: both dynamic & static mode).

default_op_dyn_en

Type

bool

Default value

0x0

Whether to enable operating mode dynamic transition by default.

default_op_policy

Type

int

Default value

0x0

Default operating policy.

default_power_state_on**Type**

bool

Default value

0x0

Default power state ON.

default_pwr_dyn_en**Type**

bool

Default value

0x0

Whether to enable dynamic power mode transition by default.

device_channels**Type**

int

Default value

0x0

Number of device channels (0: P-Channel, 1-8: Q-Channels).

dynamic_off**Type**

bool

Default value

0x0

Dynamic Off.

dynamic_on**Type**

bool

Default value

0x0

Dynamic On.

dynamic_warm_reset**Type**

bool

Default value

0x0

Dynamic Warm Reset.

full_ret**Type**

int

Default value

0x0

Full Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

func_ret**Type**

int

Default value

0x0

Functional Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

is_core_ppu**Type**

bool

Default value

0x0

PPU is core_ppu type which means wake_request would wait till PPU is OFF/OFF_EMU.

lock_support**Type**

bool

Default value

0x1

Whether to support OFF lock feature.

logic_ret**Type**

int

Default value

0x0

Logic Retention (0: not supported, 1: static, 2: dynamic).

mem_off**Type**

int

Default value

0x0

Memory Off (0: not supported, 1: static mode only, 2: both dynamic & static mode).

mem_ret**Type**

int

Default value

0x0

Memory Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

mem_ret_emu**Type**

int

Default value

0x0

Emulated Memory Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

num_opmode_cfg**Type**

int

Default value

0x0

Number of operating modes.

off_emu**Type**

int

Default value

0x0

Emulated Off (0: not supported, 1: static mode only, 2: both dynamic & static mode).

off_mem_ret_trans_cfg**Type**

bool

Default value

0x0

OFF to MEM_RET direct transition configuration (0: not allowed, 1: allowed).

op_active_cfg**Type**

int

Default value

0x0

Operating mode active configuration (0: Ladder use model, 1: Independent user model).

revision**Type**

string

Default value

"r1p1"

Revision.

use_active_signal**Type**

bool

Default value

0x0

Use device-active signal.

3.10.88 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 3-1244: 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 23.

Ports for PPUv1_Cluster_Wakerequest_Logic

Table 3-1245: 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

Set Stall Condition of Core WakeRequest (from GIC) for Core PPU after reset.

`disable_core_ppu_wakerequest_input_stall`

Type

bool

Default value

0x0

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

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.

3.10.89 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 3-1246: 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 23.

Iris and MTI instances for RSE_CPU_Private_Region

This model has the following Iris instances:

Table 3-1247: 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 3-1248: RSE_CPU_Private_Region MTI instances

InstanceName	ComponentName
RSE_CPU_Private_Region.apb_nonsecure	PVBusSlave
RSE_CPU_Private_Region.apb_secure	PVBusSlave

RSE_CPU_Private_Region contains the following CADI targets:

- RSE_CPU_Private_Region

Ports for RSE_CPU_Private_Region**Table 3-1249: Ports**

Name	Protocol	Type	Description
apb_nonsecure	PVBus	Slave	-
apb_secure	PVBus	Slave	secure & non-secure Subordinate APB Interface
reset_in	Signal	Slave	Reset in signal

Parameters for RSE_CPU_Private_Region

CPUID_RESET_VALUE

Type
int

Default value
0x0

CPUID Registers Reset Value.

diagnostics

Type
int

Default value
0x2

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.10.90 SMMUv3AEM

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1250: 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 23.

Changes in 11.27.17

Parameters added:

- mpam_sp_options

Iris and MTI instances for SMMUv3AEM

This model has the following Iris instances:

Table 3-1251: SMMUv3AEM Iris instances

InstanceName	ComponentName
SMMUv3AEM	SMMUv3AEM
SMMUv3AEM.register_file[0]	PVBusSlave
SMMUv3AEM.service_request_tbu[0]	PVBusSlave
SMMUv3AEM.service_request_tbu[10]	PVBusSlave

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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
SMMUv3AEM.tb[18]	PVBusMapper
SMMUv3AEM.tb[19]	PVBusMapper
SMMUv3AEM.tb[1]	PVBusMapper
SMMUv3AEM.tb[20]	PVBusMapper
SMMUv3AEM.tb[21]	PVBusMapper
SMMUv3AEM.tb[22]	PVBusMapper
SMMUv3AEM.tb[23]	PVBusMapper

InstanceName	ComponentName
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
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

InstanceName	ComponentName
SMMUv3AEM.tb[5]	PVBusMapper
SMMUv3AEM.tb[60]	PVBusMapper
SMMUv3AEM.tb[61]	PVBusMapper
SMMUv3AEM.tb[62]	PVBusMapper
SMMUv3AEM.tb[63]	PVBusMapper
SMMUv3AEM.tb[6]	PVBusMapper
SMMUv3AEM.tb[7]	PVBusMapper
SMMUv3AEM.tb[8]	PVBusMapper
SMMUv3AEM.tb[9]	PVBusMapper

This model has the following MTI trace components:

Table 3-1252: 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

InstanceName	ComponentName
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
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

InstanceName	ComponentName
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
SMMUv3AEM.tbu[32]	PVBusMapper
SMMUv3AEM.tbu[33]	PVBusMapper
SMMUv3AEM.tbu[34]	PVBusMapper
SMMUv3AEM.tbu[35]	PVBusMapper
SMMUv3AEM.tbu[36]	PVBusMapper
SMMUv3AEM.tbu[37]	PVBusMapper
SMMUv3AEM.tbu[38]	PVBusMapper
SMMUv3AEM.tbu[39]	PVBusMapper
SMMUv3AEM.tbu[3]	PVBusMapper
SMMUv3AEM.tbu[40]	PVBusMapper
SMMUv3AEM.tbu[41]	PVBusMapper
SMMUv3AEM.tbu[42]	PVBusMapper
SMMUv3AEM.tbu[43]	PVBusMapper

InstanceName	ComponentName
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

SMMUv3AEM contains the following CADI targets:

- SMMUv3AEM

About SMMUv3AEM

The SMMUv3 Architecture Envelope Model component is an architectural model that implements the SMMUv3.0, SMMUv3.1, and SMMUv3.2 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 TT1) then it must sign extend the address from the upstream peripherals to get to 64 bits.

The SMMUv3AEM 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 so will not be 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 then 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 model has the following limitations:

- It does not support:
 - RAS.
 - Power control.
 - AMBA® stash operations and destructive read operations are not supported on `PVBus` and also are not supported by the device.
 - PCIe-NoSnoop transactions.
- The PMU has limited functionality. Only a subset of the architecturally mandatory events are supported, as indicated by the `SMMU_PMC_G_CBEID0` fields. The PMU is intended for demonstration purposes only and for driver development.

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.
- 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`. Thus, if your system needs to distinguish MSI writes from Event queue writes, it can do so using this mechanism.
- If your system does table walks and queue accesses through TBU0 (`separate_tw_msi_qs_port == false`), then care must be taken to distinguish table walk and queue traffic (with `MasterID=0xFFFFFFFF`) 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. 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`.

Table 3-1253: Security State Determination (SSD) encoding

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

Ports for SMMUv3AEM

Table 3-1254: Ports

Name	Protocol	Type	Description
<code>axi_stream_msi_addr_to_match_s</code>	Value_64	Slave	Any SMMU-originated MSI which exactly matches the address in this port will be sent through the AXI stream port <code>axi_stream_msi_m</code> which is usually connected to the GIC through <code>axi_stream_msi_s</code> . 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 <code>smmu_msi_device_id</code> is the DeviceID to send on the interface. See also the parameters: <code>axi_stream_msi_TID</code> and <code>axi_stream_msi_TDEST</code> . The default value of this port is set by the parameter <code>axi_stream_msi_addr_to_match</code> . This port is sampled at negedge of <code>reset_in</code> and must be set before the negedge of the reset signal.
<code>axi_stream_msi_m</code>	PVBus	Master	Manager port used for sending SMMU originated MSIs directly to the GIC when <code>axi_stream_msi_enabled == true</code>
<code>clk_in</code>	ClockSignal	Slave	Clock signal
<code>conf_reset_of_SMMU_GBPA_ABORT</code>	Signal	Slave	System reset value of SMMU_GBPA.ABORT. This port is sampled at negedge of <code>reset_in</code> and must be set before the negedge of the reset signal.
<code>conf_reset_of_SMMU_S_GBPA_ABORT</code>	Signal	Slave	System reset value of SMMU_S_GBPA.ABORT. This port is sampled at negedge of <code>reset_in</code> and must be set before the negedge of the reset signal.
<code>conf_system_supports_bgptm</code>	Signal	Slave	System supports broadcast TLBI PAALL and TLBI RPA for supporting RME. This port is sampled at negedge of <code>reset_in</code> 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

If SMMU_IDR1.ATTR_PERMS_OVR == 0 then this parameter is ignored.

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 see actual transaction 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`.



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`.



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.

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_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]
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], ...
```

The `streamID` (32 b) is valid if `sidv` is 1 or both `sidv` and `nsidv` are unused.

`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].

The `substreamID` (20 b) is valid if `ssv` is true.

`SIDV == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

`nSEC_SID`, `nSSV`, `nSIDV` are available with negative logic. Different attributes are independent and can use negative or positive logic. 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`.

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]
```

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.COHAACC is a system property. However, the exact nature of the transactions that the SMMU emits is an IMP DEF property when COHAACC == 0: 0 – COHAACC == 0 forces the output attributes of SMMU-generated accesses to non-shared. 1 – The only effect of COHAACC is what is reported in SMMU_IDR0.COHAACC 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

""

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_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_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: * MPAM_SP = PAS * MPAM_PARTID = 0 * MPAM_PMG = 0

`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

""

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/MasterID/UserFlags.

RHS Symbols: * MECID * 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. * 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"
```

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/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`.

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, MasterID=0xFFFFFFFF"

Transform downstream attributes of MSI transactions.



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 was 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 rwx 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_CRO.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 `logptsz_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.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV, UserFlags[19:0]=SubstreamID,
MasterID[10]=MasterID[11], MasterID[11]=MasterID[10]"
```

RHS/LHS Symbols:

- **ExtendedID/MasterID/UserFlags**
incoming/outgoing attributes.

RHS Symbols:

- **DeviceID**
StreamID + translated_device_id_base
- StreamID/SubstreamID/SSV/SEC_SID
- **nSSV/nSEC_SID/**
negative logic versions.
- **St1PBHA/St2PBHA**
Page Based Hardware Attributes from leaf descriptors (zero if not used).
- **STE_IMPDEF1**
STE[127:116]
- **HWATTR_KIND_0**
PBHA information
- numeric literals.
- SIDV = 1, nSIDV = 0 (fixed values to indicate StreamID)

Any bits with no transform are unchanged.



- `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"

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, MasterID[10]=MasterID[11],
MasterID[11]=MasterID[10] MasterID[9:6]=HWATTR_KIND_0"
```

RHS/LHS Symbols: * ExtendedID/MasterID/UserFlags: incoming/outgoing attributes.

RHS Symbols: * SIDV = 0, nSIDV = 1 (fixed values to indicate NoStreamID) * PAS * HWATTR_KIND_0 * numeric literals.

Any bits with no transform are unchanged.



- `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"

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/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 use 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.COHAACC == 1 and SMMU_IDR0.iIDR0.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_l0gpt_entry_covers_log2size_in_bytes

Type

uint32_t

Default value

30

Each LOGPT entry covers:

```
2*rme_l0gpt_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_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_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`.

smmuv33_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.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

- ExtendedID/MasterID/UserFlags

RHS Symbols:

- HWATTR_KIND_0: PBHA information
- 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
- numeric literals.

ExtendedID/MasterID/UserFlags start with values {0, 0xFFFFffff, 0} respectively.

Any bits with no transform are unchanged.



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 & 0xFFFFfff)-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 $[0, \max(2, (t \& 0xFFFFffffff)) - 1]$.

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 $[0, \max(2, (t \& 0xFFFFffffff)) - 1]$.

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.

3.10.91 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 3-1256: 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 23.

Iris and MTI instances for SMSC_91C111

This model has the following Iris instances:

Table 3-1257: SMSC_91C111 Iris instances

InstanceName	ComponentName
SMSC_91C111	SMSC_91C111
SMSC_91C111.SMSC_slave	PVBusSlave

This model has the following MTI trace components:

Table 3-1258: SMSC_91C111 MTI instances

InstanceName	ComponentName
SMSC_91C111.SMSC_slave	PVBusSlave

SMSC_91C111 contains the following CADI targets:

- SMSC_91C111

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:

- [1.11.3 Configuring the networking environment for Microsoft Windows](#) on page 48
- [1.11.8 Configuring the networking environment for Linux](#) on page 54

Ports for SMSC_91C111

Table 3-1259: Ports

Name	Protocol	Type	Description
clock	ClockSignal	Slave	Clock input, typically 25MHz, which sets the master transmit/receive rate.
eth	VirtualEthernet	Master	Ethernet port.
intr	Signal	Master	Interrupt signal.
pvbust	PVBus	Slave	Slave port for register access.
state	ValueState_64	Master	State port to retrieve state of host bridge

Parameters for SMSC_91C111

cache_size

Type

int

Default value

0x10000

Size of cache memory in SMSC MMU.

enabled

Type

bool

Default value

0x0

Host interface connection enabled.

mac_address

Type

string

Default value

"00:02:f7:ef:00:00"

Host/model MAC address.

not_lan911x

Type

bool

Default value

0x0

Gracefully fail SMSC LAN911x driver probe.

promiscuous

Type

bool

Default value

0x1

Put host into promiscuous mode.

3.10.92 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 3-1260: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for SP804_Timer

This model has the following Iris instances:

Table 3-1261: 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 3-1262: SP804_Timer MTI instances

InstanceName	ComponentName
SP804_Timer.busslave	PVBusSlave
SP804_Timer.clk_div0	ClockDivider

InstanceName	ComponentName
SP804_Timer.clk_div1	ClockDivider

SP804_Timer contains the following CADI targets:

- ClockDivider
- CounterModule
- SP804_Timer

Ports for SP804_Timer

Table 3-1263: Ports

Name	Protocol	Type	Description
clock	ClockSignal	Slave	Clock input, typically 1MHz, driving master count rate.
irq_out0	Signal	Master	Interrupt signaling.
irq_out1	Signal	Master	Interrupt signaling.
pvbus	PVBus	Slave	Slave port for register access.
timer_en[2]	ClockRateControl	Slave	Port for changing the rate of timer n.

Parameters for SP804_Timer

`clk_div0.div`

Type

int

Default value

0x1

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

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

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

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

3.10.93 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 3-1264: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for SP805_Watchdog

This model has the following Iris instances:

Table 3-1265: 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 3-1266: SP805_Watchdog MTI instances

InstanceName	ComponentName
SP805_Watchdog.busslave	PVBusSlave

SP805_Watchdog contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- SP805_Watchdog

- SchedulerThread
- SchedulerThreadEvent

Ports for SP805_Watchdog

Table 3-1267: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock input, typically 1MHz, driving master count rate.
irq_out	Signal	Master	Interrupt signaling.
pvbus_s	PVBus	Slave	Slave port for register access.
reset_in	Signal	Slave	Reset signaling.
reset_out	Signal	Master	Reset signaling.

Parameters for SP805_Watchdog

simhalt

Type

bool

Default value

0x0

Halt on reset.

3.10.94 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 3-1268: 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 23.

Iris and MTI instances for SP810_SysCtrl

This model has the following Iris instances:

Table 3-1269: SP810_SysCtrl Iris instances

InstanceName	ComponentName
SP810_SysCtrl	SP810_SysCtrl
SP810_SysCtrl.busslave	PVBusSlave

InstanceName	ComponentName
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 3-1270: SP810_SysCtrl MTI instances

InstanceName	ComponentName
SP810_SysCtrl.busslave	PVBUSlave
SP810_SysCtrl.clkdiv_clk0	ClockDivider
SP810_SysCtrl.clkdiv_clk1	ClockDivider
SP810_SysCtrl.clkdiv_clk2	ClockDivider
SP810_SysCtrl.clkdiv_clk3	ClockDivider

SP810_SysCtrl contains the following CADI targets:

- [ClockDivider](#)
- [SP810_SysCtrl](#)

Ports for SP810_SysCtrl

Table 3-1271: 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.

Name	Protocol	Type	Description
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

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

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

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

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

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

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

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

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

sysid**Type**

int

Default value

0x0

System Identification Register.

use_s8**Type**

bool

Default value

0x0

Use Switch 8 (S1-S4).

3.10.95 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 3-1272: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for SSU

This model has the following Iris instances:

Table 3-1273: SSU Iris instances

InstanceName	ComponentName
SSU	SSU
SSU.pvbus_slave	PVBusSlave

This model has the following MTI trace components:

Table 3-1274: SSU MTI instances

InstanceName	ComponentName
SSU	SSU
SSU.pvbus_slave	PVBusSlave

SSU contains the following CADI targets:

- SSU

Ports for SSU

Table 3-1275: 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

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT:2).

sm_implemented

Type

bool

Default value

0x0

Safety Mechanism Implemented.

3.10.96 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 3-1276: 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 23.

Iris and MTI instances for SecureAlarmManager

This model has the following Iris instances:

Table 3-1277: SecureAlarmManager Iris instances

InstanceName	ComponentName
SecureAlarmManager	SecureAlarmManager
SecureAlarmManager.apb	PVBusSlave

This model has the following MTI trace components:

Table 3-1278: SecureAlarmManager MTI instances

InstanceName	ComponentName
SecureAlarmManager.apb	PVBusSlave

SecureAlarmManager contains the following CADI targets:

- SecureAlarmManager

Ports for SecureAlarmManager

Table 3-1279: 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

Number of SAM event counters.

NUM_SAMNRA

Type

int

Default value

0x7

Number of SAM response actions.

diagnostics

Type

int

Default value

0x2

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.10.97 SecureICache

SecureICache. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1280: 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 23.

Iris and MTI instances for SecureICache

This model has the following Iris instances:

Table 3-1281: 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 3-1282: 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

SecureICache contains the following CADI targets:

- SecureICache

Ports for SecureCache

Table 3-1283: 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 SecureCache

SIC_AUTH_ENABLE

Type

bool

Default value

0x0

SIC Authentication enabled [default==false].

SIC_DECRYPT_ENABLE

Type

bool

Default value

0x0

SIC Decryption enabled [default==false].

SIC_DR_CNT

Type

int

Default value

0x1

SIC config: Decryption Region Count (0x0==1, 0x1==2[default], 0x2==4).

SIC_HTR_RAM_SIZE

Type

int

Default value

0x20

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

SIC config: Maximum Code Size supported (n==nKB[default,1024(1MB)]).

SIC_PAGE_RAM_SIZE

Type

int

Default value

0x10

SIC config: Page RAM Size (0x4==4KB, 0x8==8KB[default], 0x10=16KB).

SIC_PAGE_SIZE

Type

int

Default value

0x3

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

SIC config: Performance Monitor enable [default==false].

diagnostics

Type

int

Default value

0x2

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.10.98 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 3-1284: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for SystemFMU

This model has the following Iris instances:

Table 3-1285: SystemFMU Iris instances

InstanceName	ComponentName
SystemFMU	SYSTEM_FMU
SystemFMU.pvbus_slave	PVBusSlave

This model has the following MTI trace components:

Table 3-1286: SystemFMU MTI instances

InstanceName	ComponentName
SystemFMU	SYSTEM_FMU
SystemFMU.pvbus_slave	PVBusSlave

SystemFMU contains the following CADI targets:

- SYSTEM_FMU

Ports for SystemFMU

Table 3-1287: 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

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

upstream_fmuscfg

Type

int

Default value

0x1

Number of upstream FMUs that can be connected to the System level FMU.

3.10.99 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 3-1288: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Iris and MTI instances for System_RAS_Agent

This model has the following Iris instances:

Table 3-1289: System_RAS_Agent Iris instances

InstanceName	ComponentName
System_RAS_Agent	SYSTEM_RAS_AGENT
System_RAS_Agent.PVBusSubordinate	PVBusSlave

This model has the following MTI trace components:

Table 3-1290: System_RAS_Agent MTI instances

InstanceName	ComponentName
System_RAS_Agent.PVBusSubordinate	PVBusSlave

System_RAS_Agent contains the following CADI targets:

- SYSTEM_RAS_AGENT

Ports for System_RAS_Agent

Table 3-1291: 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).
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

Number of downstream RAS agents for which the proxy error record is maintained.

SYNC_ENABLE

Type

int

Default value

0x0

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

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.10.100 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 3-1292: 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 23.**Iris and MTI instances for TZC_400**

This model has the following Iris instances:

Table 3-1293: 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 3-1294: 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

TZC_400 contains the following CADI targets:

- TZC_400

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.

Ports for TZC_400

Table 3-1295: 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

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"

Mapping from Master ID to NSAID.

master_id_from_label**Type**

bool

Default value

0x0

Obtain Master ID from label (ignores id_mapping).

rst_action**Type**

int

Default value

0x0

ACTION register value at reset.

rst_build_config**Type**

int

Default value

0x3003f08

BUILD_CONFIG register value at reset.

rst_gate_keeper**Type**

int

Default value

0x0

GATE_KEEPER register value at reset.

rst_region_attributes_0**Type**

int

Default value

0xf

Region 0 Secure attributes.

rst_region_attributes_1**Type**

int

Default value

0x0

Region 1 Secure attributes.

rst_region_attributes_2**Type**

int

Default value

0x0

Region 2 Secure attributes.

rst_region_attributes_3**Type**

int

Default value

0x0

Region 3 Secure attributes.

rst_region_attributes_4**Type**

int

Default value

0x0

Region 4 Secure attributes.

rst_region_attributes_5**Type**

int

Default value

0x0

Region 5 Secure attributes.

rst_region_attributes_6**Type**

int

Default value

0x0

Region 6 Secure attributes.

rst_region_attributes_7**Type**

int

Default value

0x0

Region 7 Secure attributes.

rst_region_attributes_8**Type**

int

Default value

0x0

Region 8 Secure attributes.

rst_region_base_high_1**Type**

int

Default value

0x0

Region 1 base memory address (high 32 bits).

rst_region_base_high_2**Type**

int

Default value

0x0

Region 2 base memory address (high 32 bits).

rst_region_base_high_3**Type**

int

Default value

0x0

Region 3 base memory address (high 32 bits).

rst_region_base_high_4**Type**

int

Default value

0x0

Region 4 base memory address (high 32 bits).

rst_region_base_high_5**Type**

int

Default value

0x0

Region 5 base memory address (high 32 bits).

rst_region_base_high_6**Type**

int

Default value

0x0

Region 6 base memory address (high 32 bits).

rst_region_base_high_7**Type**

int

Default value

0x0

Region 7 base memory address (high 32 bits).

rst_region_base_high_8**Type**

int

Default value

0x0

Region 8 base memory address (high 32 bits).

rst_region_base_low_1**Type**

int

Default value

0x0

Region 1 base memory address (low 32 bits).

rst_region_base_low_2**Type**

int

Default value

0x0

Region 2 base memory address (low 32 bits).

rst_region_base_low_3**Type**

int

Default value

0x0

Region 3 base memory address (low 32 bits).

rst_region_base_low_4**Type**

int

Default value

0x0

Region 4 base memory address (low 32 bits).

rst_region_base_low_5**Type**

int

Default value

0x0

Region 5 base memory address (low 32 bits).

rst_region_base_low_6**Type**

int

Default value

0x0

Region 6 base memory address (low 32 bits).

rst_region_base_low_7**Type**

int

Default value

0x0

Region 7 base memory address (low 32 bits).

rst_region_base_low_8**Type**

int

Default value

0x0

Region 8 base memory address (low 32 bits).

rst_region_id_access_0**Type**

int

Default value

0x0

Region 0 NSAID permissions.

rst_region_id_access_1**Type**

int

Default value

0x0

Region 1 NSAID permissions.

rst_region_id_access_2**Type**

int

Default value

0x0

Region 2 NSAID permissions.

rst_region_id_access_3**Type**

int

Default value

0x0

Region 3 NSAID permissions.

rst_region_id_access_4**Type**

int

Default value

0x0

Region 4 NSAID permissions.

rst_region_id_access_5**Type**

int

Default value

0x0

Region 5 NSAID permissions.

rst_region_id_access_6**Type**

int

Default value

0x0

Region 6 NSAID permissions.

rst_region_id_access_7**Type**

int

Default value

0x0

Region 7 NSAID permissions.

rst_region_id_access_8**Type**

int

Default value

0x0

Region 8 NSAID permissions.

rst_region_top_high_0**Type**

int

Default value

0x0

Region 0 (default) top memory address.

rst_region_top_high_1**Type**

int

Default value

0x0

Region 1 top memory address (high 32 bits).

rst_region_top_high_2**Type**

int

Default value

0x0

Region 2 top memory address (high 32 bits).

rst_region_top_high_3**Type**

int

Default value

0x0

Region 3 top memory address (high 32 bits).

rst_region_top_high_4**Type**

int

Default value

0x0

Region 4 top memory address (high 32 bits).

rst_region_top_high_5**Type**

int

Default value

0x0

Region 5 top memory address (high 32 bits).

rst_region_top_high_6**Type**

int

Default value

0x0

Region 6 top memory address (high 32 bits).

rst_region_top_high_7**Type**

int

Default value

0x0

Region 7 top memory address (high 32 bits).

rst_region_top_high_8**Type**

int

Default value

0x0

Region 8 top memory address (high 32 bits).

rst_region_top_low_1**Type**

int

Default value

0x0

Region 1 top memory address (low 32 bits).

rst_region_top_low_2**Type**

int

Default value

0x0

Region 2 top memory address (low 32 bits).

rst_region_top_low_3**Type**

int

Default value

0x0

Region 3 top memory address (low 32 bits).

rst_region_top_low_4**Type**

int

Default value

0x0

Region 4 top memory address (low 32 bits).

rst_region_top_low_5**Type**

int

Default value

0x0

Region 5 top memory address (low 32 bits).

rst_region_top_low_6**Type**

int

Default value

0x0

Region 6 top memory address (low 32 bits).

rst_region_top_low_7**Type**

int

Default value

0x0

Region 7 top memory address (low 32 bits).

rst_region_top_low_8**Type**

int

Default value

0x0

Region 8 top memory address (low 32 bits).

3.10.101 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 3-1296: 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 23.

Iris and MTI instances for TZFilterUnit

This model has the following Iris instances:

Table 3-1297: TZFilterUnit Iris instances

InstanceName	ComponentName
TZFilterUnit	TZFilterUnit
TZFilterUnit.BusMapper	PVBusMapper

This model has the following MTI trace components:

Table 3-1298: TZFilterUnit MTI instances

InstanceName	ComponentName
TZFilterUnit.BusMapper	PVBusMapper

TZFilterUnit contains the following CADI targets:

- [TZFilterUnit](#)

Ports for TZFilterUnit

Table 3-1299: Ports

Name	Protocol	Type	Description
control	TZFilterControl	Master	Configuration port.
pvbus_m	PVBus	Master	Master bus port.
pvbus_s	PVBus	Slave	Slave bus port.

3.10.102 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 3-1300: 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 23.

Iris and MTI instances for TZIC

This model has the following Iris instances:

Table 3-1301: TZIC Iris instances

InstanceName	ComponentName
TZIC	TZIC
TZIC.busslave	PVBusSlave

This model has the following MTI trace components:

Table 3-1302: TZIC MTI instances

InstanceName	ComponentName
TZIC.busslave	PVBusSlave

TZIC contains the following CADI targets:

- TZIC

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.

Ports for TZIC

Table 3-1303: 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.
pvbuss	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.

3.10.103 TrustedRAM

Trusted RAM. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1304: 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 23.

Iris and MTI instances for TrustedRAM

This model has the following Iris instances:

Table 3-1305: TrustedRAM Iris instances

InstanceName	ComponentName
TrustedRAM	TrustedRAM
TrustedRAM.apb	PVBusSlave

This model has the following MTI trace components:

Table 3-1306: TrustedRAM MTI instances

InstanceName	ComponentName
TrustedRAM.apb	PVBusSlave

TrustedRAM contains the following CADI targets:

- TrustedRAM

Ports for TrustedRAM

Table 3-1307: 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

TRBC Registers Reset Value.

diagnostics

Type

int

Default value
0x2

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.10.104 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 3-1308: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Ports for VHT_VIOBridge

Table 3-1309: 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
""

Baseline of scripts to use for VIO.

vio_path

Type
string

Default value
""

Path to find python scripts for VIO.

3.10.105 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 3-1310: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Ports for VHT_VSIBridge

Table 3-1311: Ports

Name	Protocol	Type	Description
intr	Signal	Master	Interrupt raising signal
pvbus_m	PVBus	Master	Requester port for access to emeory
pvbus_s	PVBus	Slave	Completer port for access to VSIBridge registers

Parameters for VHT_VSIBridge

vsi_basename

Type

string

Default value

""

Baseline of scripts to use for VSI.

vsi_idx

Type

int

Default value

0x0

Index to use for VSI.

vsi_path

Type

string

Default value

""

Path to find python scripts for VSI.

3.10.106 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 3-1312: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 23.

Ports for VHT_VSocket

Table 3-1313: 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

""

Name of the component.

3.10.107 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 3-1314: 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 23.

Iris and MTI instances for v7_VGIC

This model has the following Iris instances:

Table 3-1315: 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 3-1316: v7_VGIC MTI instances

InstanceName	ComponentName
v7_VGIC	v7_VGIC
v7_VGIC.vgic_bus_slave	PVBusSlave

v7_VGIC contains the following CADI targets:

- v7_VGIC

Ports for v7_VGIC

Table 3-1317: 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.

Name	Protocol	Type	Description
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

Implementation ID to present for the cores.

dist-impl-id

Type

int

Default value

0x3902043b

Implementation ID to present for the distributor.

enable_log_errors

Type

bool

Default value

0x0

.

enable_log_fatal

Type

bool

Default value

0x0

.

enable_log_warnings

Type

bool

Default value

0x0

enabled**Type**

bool

Default value

0x1

Enable the component. If it is disabled then all register writes will have no effect.

number-of-cores**Type**

int

Default value

0x8

Number of core interfaces to present.

number-of-ints**Type**

int

Default value

0xe0

Number of interrupt pins. Will be rounded down to the nearest multiple of 32.

number-of-lrs**Type**

int

Default value

0x40

Number of list registers.

vgic-version**Type**

int

Default value

0xb

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
```

Related information

- [scx::scx_parse_and_configure](#)

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-7.3/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
```



- 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 4754

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](#).



- 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	false	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_A55x2.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`.

4.5 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:

```
./EVS_Base_Cortex-A73x1 -a __image.axf \
...
--plugin $PVLIB_HOME/plugins/Linux64_GCC-7.3/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:

```
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-7.3/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.

Related information

- [Connect Model Debugger to a model running on another host](#)

4.6.1 CADIIPCRemoteConnection - parameters

This section describes the parameters for the CADIIPCRemoteConnection plug-in.

Each parameter is prefixed with `REMOTE_CONNECTION.CADIIPCRemoteConnection`, for example:

```
REMOTE_CONNECTION.CADIIPCRemoteConnection.enable_remote_cadi
```

Table 4-4: CADIIPCRemoteConnection parameters

Name	Type	Default value	Allowed values	When set	Description
<code>enable_remote_cadi</code>	bool	false	true, false	Init time	Allow connections from remote hosts.
<code>listen_address</code>	string	"127.0.0.1"	""	Init time	If <code>enable_remote_cadi</code> is set, the network address the server listens on. The default is 127.0.0.1.
<code>port</code>	int	0x7b8b	0x1 - 0xffff	Init time	If <code>enable_remote_cadi</code> is set, the TCP port the server listens on. The default is 31627.
<code>range</code>	int	0x0	0x0 - 0x64	Init time	If the requested port is not available, search for the next available port in the range [port:port+range]. The default is zero, which means only try the requested port.

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 CADI 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 `d` 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 4-5: CDETester parameters

Name	Type	Default value	Allowed values	When set	Description
<code>cde_tester_trivial.cps_implemented</code>	int	<code>0xff00ff</code>	<code>0x0 - 0xffffffff</code>	Init time	Bitmask indicating coprocessors implemented.
<code>cde_tester_trivial.cps_implemented_cx1</code>	int	<code>0xff00ff</code>	<code>0x0 - 0xffffffff</code>	Init time	Coprocessor enables CX1 instructions ([23:16] for CX1A, [7:0] for CX1).
<code>cde_tester_trivial.cps_implemented_cx1d</code>	int	<code>0xff00ff</code>	<code>0x0 - 0xffffffff</code>	Init time	Coprocessor enables CX1D instructions ([23:16] for CX1DA, [7:0] for CX1D).
<code>cde_tester_trivial.cps_implemented_cx2</code>	int	<code>0xff00ff</code>	<code>0x0 - 0xffffffff</code>	Init time	Coprocessor enables CX2 instructions ([23:16] for CX2A, [7:0] for CX2).

Name	Type	Default value	Allowed values	When set	Description
cde_tester_trivial.cps_implemented_cx2d	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX2D instructions ([23:16] for CX2DA, [7:0] for CX2D).
cde_tester_trivial.cps_implemented_cx3	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX3 instructions ([23:16] for CX3A, [7:0] for CX3).
cde_tester_trivial.cps_implemented_cx3d	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX3D instructions ([23:16] for CX3DA, [7:0] for CX3D).
cde_tester_trivial.cps_implemented_vcx1	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX1 instructions ([23:16] for VCX1A, [7:0] for VCX1).
cde_tester_trivial.cps_implemented_vcx1v	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX1 (Vector) instructions ([23:16] for VCX1A (Vector), [7:0] for VCX1 (Vector)).
cde_tester_trivial.cps_implemented_vcx2	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX2 instructions ([23:16] for VCX2A, [7:0] for VCX2).
cde_tester_trivial.cps_implemented_vcx2v	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX2 (Vector) instructions ([23:16] for VCX2A (Vector), [7:0] for VCX2 (Vector)).
cde_tester_trivial.cps_implemented_vcx3	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX3 instructions ([23:16] for VCX3A, [7:0] for VCX3).
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 4-6: 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 CADI 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 of parameters passed through the command line or CADI.
- 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 is 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 CADI and MTI-related 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.

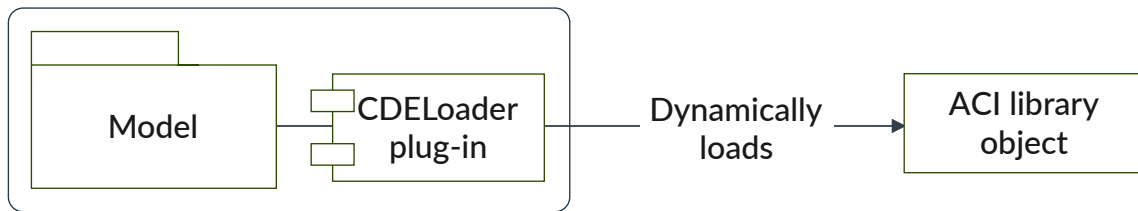
As the CDELoader framework uses a C99 API, the library can be written in C or in another language that can produce ABI-compatible shared objects and can interface with C. For example, C++, Rust, Go, or Python, through C extensions.



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 4-7: 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 4-8: Mapping CDE instructions to ACI library function definitions

Instruction	ACILibraryAPI.h function
cx1	aci_exec_cx1()
cx2	aci_exec_cx2()
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()



Note

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, `aci_handle`, 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 4-9: Crypto parameters

Name	Type	Default value	Allowed values	When set	Description
authentication_algorithm	string	"QARMA5"	""	Init time	Choice of PACAlgorithm. Valid values: "QARMA5", "QARMA3". "QARMA3" can be enabled only if the core feature has_qarma3_pac 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 .
generic_authentication_core_pattern	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 .
pointer_authentication_core_pattern	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 .
verbose	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 4-10: 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.

Name	Type	Default value	Allowed values	When set	Description
trace-source	string	"INST"	""	Init time	The trace source to be counted. Example: BRA_DIR.

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 console.

Specify one or more trace sources using the `trace-sources` parameter, for example:

```
./FVP_Base_AEMvA \
--plugin $PVLIB_HOME/plugins/Linux64_GCC-7.3/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=CACHE_*
```

- To trace a specific component only, specify the hierarchical path to it, optionally with wildcards, for example:

```
TRACE.GenericTrace.trace-sources=FVP_Base_AEMvA.cluster0.cpu1.*
```

- 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=0xFFFF7F
```

- 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 4-11: 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.
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.
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.

Name	Type	Default value	Allowed values	When set	Description
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 - 0x7fffffffffffffffff	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_IrqGpuControl

ID, name, and state of the IRQ signal from the GPU. The state can be `y` for `set`, or `n` for `clear`.

INFO_IrqJobControl

ID, name, and state of the IRQ signal from the Job Manager on the GPU. The state can be `y` for set, or `n` for clear.

INFO_IrqMmuControl

ID, name, and state of the IRQ signal from the MMU on the GPU. 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.

- 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=0x0000000000001870
VALUE=0x00000000 UPDATED_VALUE=0x00000000 REG_NAME="JOB_SLOT0_JS_FLUSH_ID_NEXT"
HOST_TIME=1515565849.691304s gpu.INFO_WriteRegister: REG_OFFSET=0x0000000000001860
VALUE=0x00000000 UPDATED_VALUE=0x00000001 REG_NAME="JOB_SLOT0_JS_COMMAND_NEXT"
HOST_TIME=1515565849.691322s gpu.INFO_IrqJobControl: IRQ_ID=0x01 IRQ_NAME="JOB
Control" IRQ_STATE=Y
HOST_TIME=1515565849.691561s gpu.INFO_ReadRegister: REG_OFFSET=0x000000000000100c
VALUE=0x00000001 REG_NAME="JOB_IRQ_STATUS"
HOST_TIME=1515565849.691643s gpu.INFO_WriteRegister: REG_OFFSET=0x0000000000001004
VALUE=0x00000000 UPDATED_VALUE=0x00000001 REG_NAME="JOB_IRQ_CLEAR"
HOST_TIME=1515565849.691647s gpu.INFO_IrqJobControl: IRQ_ID=0x01 IRQ_NAME="JOB
Control" 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 4-12: 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 libete-plugin

The libete-plugin enables the Embedded Trace Extension (ETE) for AEMvA and applicable CPU implementations.



Note

ETE support is integrated into the CT models. We recommend you configure ETE using the model parameters that are prefixed with `ete.`, which are equivalent to the plug-in parameters, instead of using this plug-in. The plug-in is deprecated and will be removed in a future release.

4.13.1 libete-plugin - parameters

This section describes the parameters for the libete-plugin plug-in.

Each parameter is prefixed with `TRACE.libete-plugin`, for example:

```
TRACE.libete-plugin.ASYNC_PACKETS_WHEN_VIEWINST_OFF
```

Table 4-13: libete-plugin parameters

Name	Type	Default value	Allowed values	When set	Description
ASYNC_PACKETS_WHEN_VIEWINST_OFF	bool	false	true, false	Init time	Generate the non-periodic alignment synchronisation packet generation when trace unit is operative.
ATBTRIG	bool	true	true, false	Init time	ATB trigger support.
CCITMIN	int	0x4	0x0 - 0xffff	Init time	Minimum cycle count value.
CCSIZE	int	0xc	0xc - 0x14	Init time	Cycle counter size.
CLAIMTAGS	int	0x8	0x0 - 0x20	Init time	Number of claim tags.
COMMOPT	bool	true	true, false	Init time	Commit mode.
COMMTRANS	bool	false	true, false	Init time	Commit transaction mode.
DEBUG	int	0x2	0x0 - 0xf	Runtime	DEBUG.
DESIGNER	int	0x41	0x0 - 0xff	Init time	DESIGNER value.
ETE_REVISION	int	0x0	0x0 - 0x3	Init time	ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.
EXCEPTION_WITH_CONTEXT	bool	true	true, false	Init time	Whether EXCEPTION_WITH_CONTEXT packet is supported.
EXPLICITLY_COMMIT_PO_ELEMS	bool	false	true, false	Init time	Whether to unilaterally explicitly emit a commit after a PO packet.
IMPDEFEXCEPPERCENTAGE	int	0x0	0x0 - 0x32	Init time	Percentage of IMPDEF exceptions inserted in instruction blocks.
IMPDEF_TRACE_ON	int	0x0	0x0 - 0x3	Init time	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.
IMPRECISE_FILTERING	int	0x0	0x0 - 0x14	Init time	Number of instruction blocks traced on a transition in the filtering.

Name	Type	Default value	Allowed values	When set	Description
LPOVERRIDE	bool	true	true, false	Init time	Low power override.
MAXSPEC	int	0x0	0x0 - 0xffffffff	Init time	Maximum speculation depth.
MAX_INST_PER_Q	int	0x1	0x1 - 0x1000	Init time	Maximum limit for the number of instructions implied by a Q element.
NOOVERFLOW	bool	false	true, false	Init time	No overflow.
NUMACPAIRS	int	0x4	0x0 - 0x8	Init time	Number of instruction address comparators pairs.
NUMCIDC	int	0x1	0x0 - 0x8	Init time	Number of context ID comparators.
NUMCNTR	int	0x2	0x0 - 0x4	Init time	Number of counters.
NUMEXTINSEL	int	0x4	0x0 - 0x4	Init time	Number of external input selectors.
NUMPC	int	0x0	0x0 - 0x8	Init time	Number of PE comparators.
NUMSEQSTATE	int	0x4	0x0 - 0x4	Init time	Number of sequencer states.
NUMSSCC	int	0x1	0x0 - 0x8	Init time	Number of single shot comparators.
NUMVMIDC	int	0x1	0x0 - 0x8	Init time	Number of virtual ID comparators.
NumberOfETEEEvents	int	0x2	0x0 - 0x4	Init time	Number of trace events.
NumberOfRSPairs	int	0x8	0x0 - 0x10	Init time	Number of resource selector pairs.
PIDR_CM0D	int	0x0	0x0 - 0xf	Init time	TRCPIDR CM0D value.
PIDR_DESIGNER	int	0x0	0x0 - 0x7ff	Init time	TRCPIDR DESIGNER value.
PIDR_PART	int	0x0	0x0 - 0xfff	Init time	TRCPIDR PART number value.
PIDR_REVAND	int	0x0	0x0 - 0xf	Init time	TRCPIDR REVAND value.
PIDR_REVISION	int	0x0	0x0 - 0xf	Init time	TRCPIDR REVISION value.
QFILT	bool	false	true, false	Init time	Q filtering.
QSUP	int	0x0	0x0 - 0x3	Init time	Q support.
Q_CADENCE	int	0x1	0x1 - 0x1000	Init time	Number of instruction blocks traced between two Q elements.
RAZWI_REG_SEL_TOP_BIT	bool	false	true, false	Init time	Implement Resource Selectors or Resource Selector Pairs bits as RAZ/WI.
REG_ACCESS_ONLY_MODE	bool	false	true, false	Init time	If enabled, all traces are disabled. Plugin only allows register acceses.
RES0_STATEFUL	bool	false	true, false	Init time	Whether RES0 bits are stateful or RAZ/WI.
RETSTACK	int	0x3	0x0 - 0xf	Init time	Return stack depth.
REVISION	int	0x0	0x0 - 0xf	Init time	TRCIDR1 revision value.
SIM_OVERFLOW_GRANULARITY	int	0x64	0xa - 0xffffffff	Init time	Number of instruction blocks in each granule, for simulated overflow.
SIM_OVERFLOW_PERCENTAGE	int	0x0	0x0 - 0x63	Init time	Percentage of instruction blocks lost in each granule, for simulated overflow.
SOURCE_ADDRESS	bool	false	true, false	Init time	Allow generation of source address elements.
STALLCTRL	bool	true	true, false	Init time	Stall control.
SYSSTALL	bool	true	true, false	Init time	System stall.
TRACEIDSIZE	int	0x7	0x0 - 0x7	Init time	Trace ID size.
TRACE_OUTPUT	string	""	""	Init time	File to which to write trace byte stream.

Name	Type	Default value	Allowed values	When set	Description
TRACE_OUTPUT_ENABLE	bool	false	true, false	Init time	ETE Trace output enable: 1=enable, 0=disable.
TRCRSRTA_FORCED_EXCEP	bool	false	true, false	Init time	TRCRSR.TA value for a forcibly traced exception.
TSMARK	bool	false	true, false	Init time	Whether timestamp markers are supported.
TSSIZE	int	0x8	0x8 - 0x8	Init time	Timestamp size.
WFXMODE	bool	true	true, false	Init time	WFX mode.

4.14 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/.

4.14.1 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 4-14: 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.

Name	Type	Default value	Allowed values	When set	Description
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.

4.15 PipelineModel

Use the `PipelineModel` plug-in to estimate the performance of workloads within a Fast Models environment.

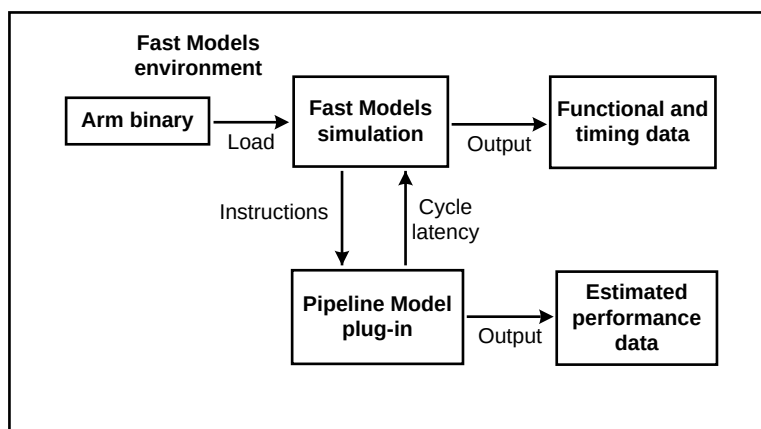


The `PipelineModel` plug-in is deprecated. It might be modified or removed in a future release.

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.



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.

4.15.1 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 4-15: 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.

4.15.2 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 4-16: InOrderPipelineModel parameters

Name	Type	Default value	Allowed values	When set	Description
core-type	string	""	""	Init time	Core type description.
instance-name	string	""	""	Init time	Cluster and core num description.

4.15.3 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.

4.15.4 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
```

4.15.5 Example command lines

The following command lines show how to load the `PipelineModel` plug-in with a variety of platforms. Any parameters that are not relevant to the examples have been omitted.

Default usage, with no options

The following command line attaches the plug-in to any instance and sends output to `stdout`:

```
./EVS_Base_Cortex-A53x1.x
...
--plugin CortexA53PipelineModel.so
```

Single cluster, single processor platform

The following command line attaches the plug-in to `cluster0.cpu0`, which is a Cortex®-A53 processor. It outputs the file `stat.txt`:

```
./EVS_Base_Cortex-A53x1.x
...
--plugin CortexA53PipelineModel.so
-C PipelineModel.CortexA53PipelineModel.core-type=ARM_Cortex-A53
-C PipelineModel.CortexA53PipelineModel.instance-name=Base.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:

```
./EVS_Base_Cortex-A53x2.x
...
--plugin 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 `cluster0.cpu0`. The second plug-in instance is named `cortexA53_1` and is attached to `cluster0.cpu1`:

```
./EVS_Base_Cortex-A53x2.x
...
--plugin CortexA53_0=CortexA53PipelineModel.so
--plugin CortexA53_1=CortexA53PipelineModel.so
-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
```

Multiclustler, single processor platform

The following command line names the plug-in instance `core1` and attaches it to `cluster1.cpu0`:

```
./EVS_Base_Cortex-A73x1-A53x1.x
...
--plugin Core1=CortexA53PipelineModel.so
-C PipelineModel.Core1.core-type=ARM_Cortex-A53
-C PipelineModel.Core1.instance-name=Base.cluster1.cpu0
```

Plug-in core-type mismatch

The following command line specifies a Cortex®-A55 processor. This mismatches the platform, which is Cortex®-A53. The plug-in will fail to load:

```
./EVS_Base_Cortex-A53x1.x
...
--plugin 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 will fail to load:

```
./EVS_Base_Cortex-A53x1.x
...
--plugin CortexA53PipelineModel.so
-C PipelineModel.CortexA53PipelineModel.instance-name=Base.cluster0.cpu2
```

Range processing

The following command line specifies 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. This is the same as not specifying a range at all:

```
-C PipelineModel.CortexA53PipelineModel.start-pc=0x80000001
-C PipelineModel.CortexA53PipelineModel.end-pc=0x80000001
```

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 causes the `PipelineModel` to process all instructions until the specified cycle limit is reached:

```
./EVS_Base_Cortex-A53x1.x
...
--cyclelimit=30000000
--plugin 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 has been reached and the `start-pc` has not yet been hit, the `PipelineModel` does not process any instructions.

4.15.6 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.

4.16 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/`.

4.17 Sidechannel

The Sidechannel plug-in allows communication between the software on the host and software on the target. It is no longer used.

4.18 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`.



The platform name is trimmed from the component name.

4.18.1 TarmacText - parameters

This section describes the parameters for the TarmacText plug-in.

Each parameter is prefixed with `TRACE.TarmacText`, for example:

```
TRACE.TarmacText.component
```

Table 4-17: TarmacText parameters

Name	Type	Default value	Allowed values	When set	Description
<code>component</code>	string	<code>"*"</code>	""	Init time	A space separated list of component to trace. Supports globbing (see man 7 glob). 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 log parameter as often as possible. Decrease the performances.
<code>log</code>	string	<code>"tarmac.@COMP@.log"</code>	""	Init time	Filename to log the text tarmac 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 performances until the portion of interest is reached. Does not impact the ExecStep Iris EventSource.

4.19 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](#).

4.19.1 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 4-18: TarmacTrace parameters

Name	Type	Default value	Allowed values	When set	Description
end-instruction-count	int	0x0	0x0 - 0x7fffffffffffffff	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 - 0x7fffffff	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 - 0x7fffffffffffffff	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_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.
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_gpt	bool	true	true, false	Init time	Trace packets generated by the GPT.
trace_hacdbbs	bool	true	true, false	Init time	Trace packets generated by the HACDBS.

Name	Type	Default value	Allowed values	When set	Description
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.

4.19.2 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><psecurity>}
```

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>.

<psecurity>

either **_ns** if the security regime of the physical address is Non-secure, or not present if the regime is Secure.



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.

4.19.3 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 SCTL_EL1,x0
98 clk R SCTL_EL1 00000000:00001005
98 clk TTW ITLB LPAE 1:1 000080002010 00000000080000705 : BLOCK ATTRIDX=1 NS=0 AP=0
SH=3 AF=1 nG=0 16E=0 PXN=0 XN=0 ADDR=0x0000000008000000
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
...
```

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 0 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 0 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 LPAAE 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.

4.19.4 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.

4.19.5 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

4.19.6 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**, **FPSCR**, 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>

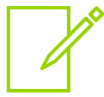
Unit for `<time>`. `ps` means simulation time, `clk` means instruction count.

<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.

4.19.7 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>
<mementype>}
```


<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.

4.19.8 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.

4.19.9 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 LPAE format.

BLOCK {<attr>=<value>}

LPAE 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.

LARGE PAGE {<attr>=<value>}

VMSA region descriptor.

SMALLPAGE {<attr>=<value>}

VMSA region descriptor.

4.19.10 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.

4.19.11 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>
{<memtype>} {<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.

4.19.12 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 4-19: Supported values for value, number, and desc

Number	Event description	Value
0x00000001	CoreEvent_Reset	-
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	-

Number	Event description	Value
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	-

4.19.13 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.



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).

4.19.14 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).

4.19.15 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<wrcbs>.

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**

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.

4.20 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

4.20.1 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 4-20: 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.

4.20.2 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 Model Debugger 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
CADI	Example C++ applications that demonstrate how to use the CADI debug API. Note: CADI is deprecated and will be replaced by the Iris debug API.
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. The following examples are also available as pre-built libraries under <code>\$PVLIB_HOME/plugins</code> : <ul style="list-style-type: none"> GenericCounter. GenericTrace. ListTraceSources.
SystemCExport	<ul style="list-style-type: none"> Source code and makefiles for EVS platform examples and SVPs. LISA+ source for bridges and EVS components. Header files required for exporting LISA+ protocols to SystemC.



Note

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 CADI examples

Example CADI clients that demonstrate how to use the CADI API to perform common debugger operations. These include instantiating a new simulation, connecting to a running simulation, accessing memory and registers, setting breakpoints, and registering callbacks for asynchronous feedback from the simulation.

Each of these examples has a `readme.txt` that describes the example, how to run it, and in some cases, gives the expected output. The following CADI examples are provided:

Table 5-2: CADI examples

Example	Description
breakpoints	Sets a code breakpoint at a specific address in the application that is running on the model.
cache_dump	Displays information about the caches in the running simulation.

Example	Description
<code>cadi_lib</code>	Implements a shared library that contains a CADI simulation. You can load the library into Model Shell and connect a CADI-compliant debugger to it.
<code>cadi_server</code>	Implements an executable that contains a CADI simulation. It loads the plug-in library <code>CADIIPCRemoteConnection.so</code> , which enables you to connect a CADI-compliant debugger to it. You can run the <code>connecting_running_target</code> CADI example to mimic this connection.
<code>connecting_instantiating_model</code>	Instantiates a simulation that is a shared library, prints a list of all targets in the simulation, and connects to one of them.
<code>connecting_remote_target</code>	Connects to a remote simulation that has been configured to allow remote connections. The IP address and listening port are passed as parameters to the CADI client.
<code>connecting_running_target</code>	Connects to a running CADI target.
<code>disassembly</code>	Prints the disassembly of the instruction at the PC address in the application that is running on the model.
<code>memory</code>	Displays memory space information. Reads and displays some values from memory in the application that is running on the model.
<code>mti</code>	Loads the TarmacTrace MTI plug-in into the application that is running on the model and displays some tarmac trace output.
<code>registers</code>	Reads and displays register information for a CPU target in the model and reads and writes a specific register.
<code>tlb_dump</code>	Connects to a model and displays the contents of the TLBs.

Related information

[Component Architecture Debug Interface User Guide](#)

5.2 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 5-3: LISA examples

Example	Description
<code>BusComponents</code>	Example LISA+ components that demonstrate different ways of using the <code>PVBus</code> interface.
<code>Common</code>	FVP-specific LISA+ components that are common to different types of FVPs.
<code>CSS</code>	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 Arm Ecosystem FVPs for more information.

Example	Description
FVP_Base	Source and project files for Base Platform FVP examples. For information about the Base Platform, see Base Platform .
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 Base Platform RevC .
FVP_Coproc_Demo	Example implementation of the <code>Coprocessor</code> interface. Registers the coprocessor with a ARMCortexM33CT or ARMAEMv8MCT model. For more information, see CoprocBusProtocol protocol .
FVP_MPS2	Source and project files for MPS2-based example platforms. For information about the MPS2 platforms, see Microcontroller Prototyping System 2 .
FVP_MPS3	Source and project files for MPS3-based example platforms that support the Arm®Corstone™ SSE-300 Example Subsystem. For more information, see MPS2 Platform FVPs in the FVP Reference Guide and Arm Corstone SSE-300 Example Subsystem Technical Reference Manual .
FVP_VE	Source and project files for VE FVPs. For information about the VE platform, see Versatile Express Model .
VP_PChannel	Shows how to create power controllers to control the power state of the cores and cluster, using the <code>PChannel</code> protocol. For information about <code>PChannel</code> , see PChannel protocol .

5.3 Build and run an FVP example

The FVP examples are located under `$PVLIB_HOME/examples/LISA/.f ### Before you begin`
{.section}

About this task

Follow these instructions to build and run one of the Base Platform FVP examples.

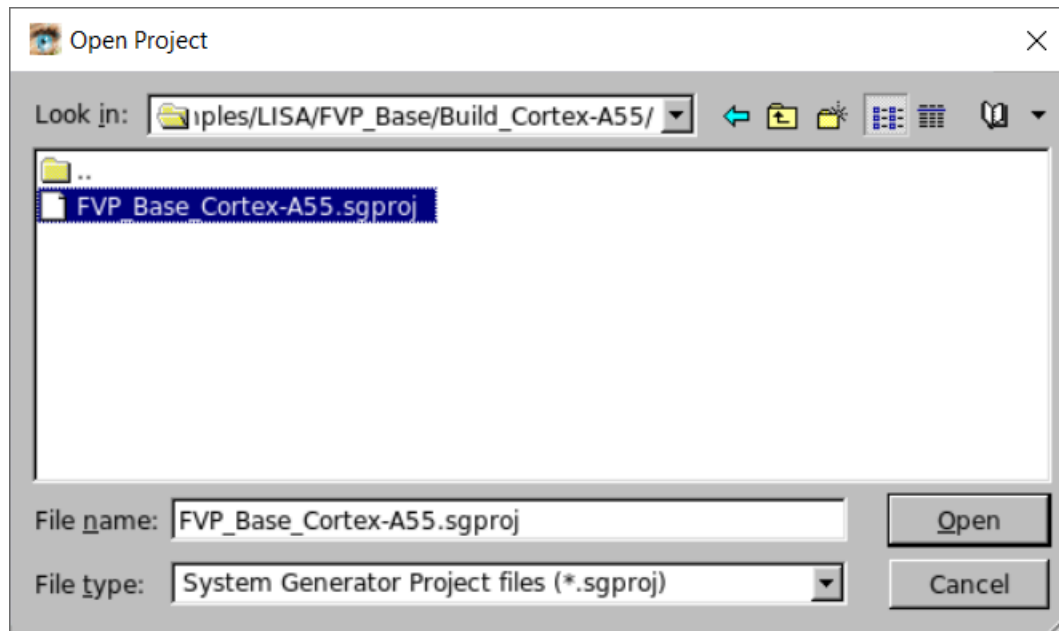


Note

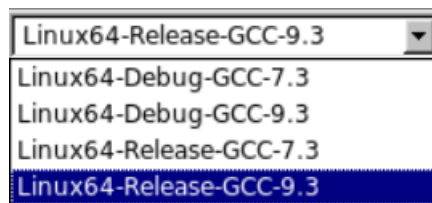
These examples build an executable platform model, which is sometimes referred to as an ISIM (Integrated SIMulator).

Procedure

1. These examples are built using System Canvas from a System Generator (SimGen) project file, with a `.sgproj` extension. To start System Canvas, open a terminal and type `sgcanvas`.
2. 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

- When the project has finished loading, select the build configuration from the Active Project Configuration drop-down menu on the main toolbar:

Figure 5-2: Select Active Project Configuration menu

- 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**.
- 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`.
- You can run `isim_system` either from the terminal or from within System Canvas:
 - To run `isim_system` from the terminal:
 - Navigate to the directory where it is located.
 - To see a full list of command-line options for `isim_system`, run it with the `--help` option:

```
./isim_system --help
```

- 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

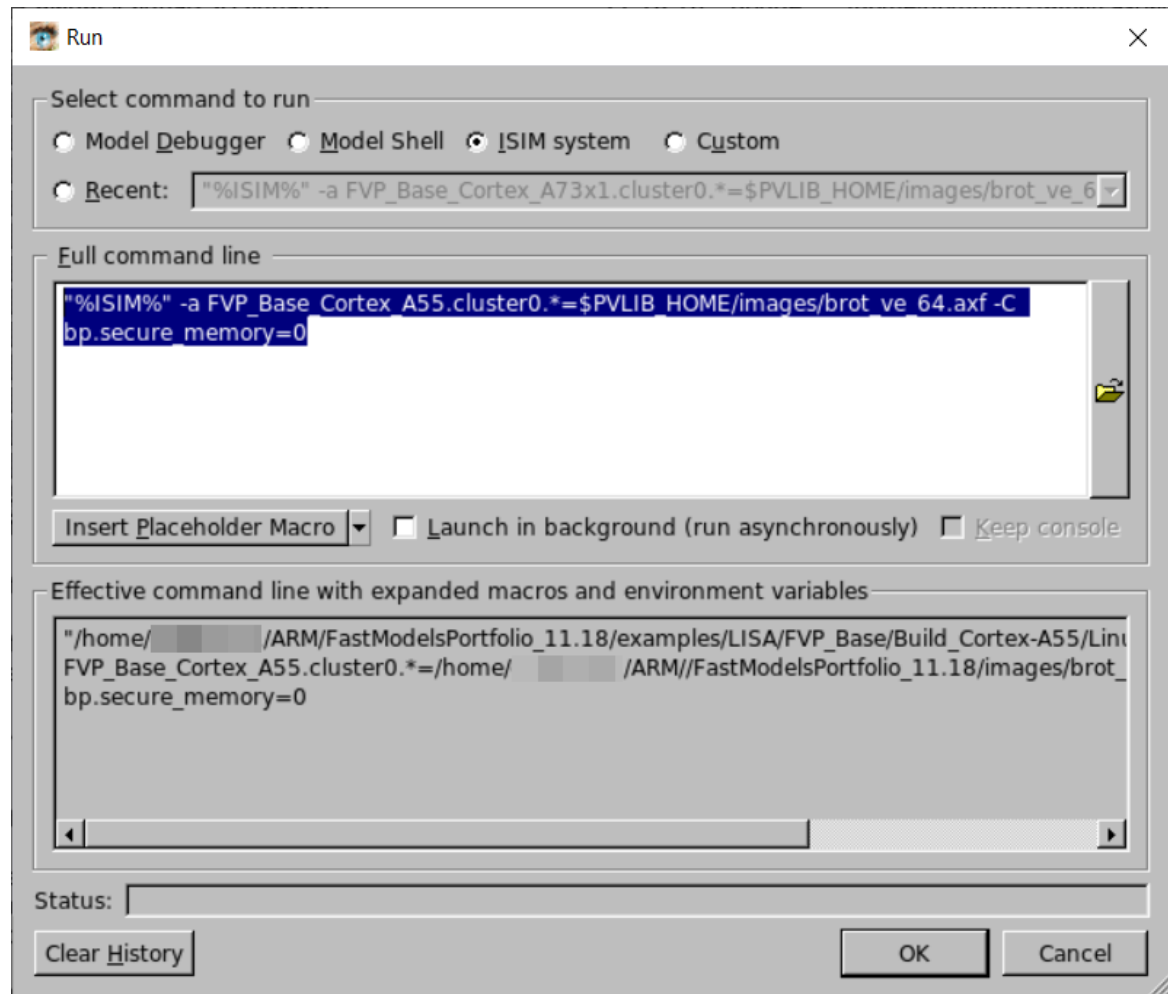
is the name of the application to load. Optionally it also specifies which core instances to run it on, in this case all cores in cluster0.

-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>`.

- The ISIM starts running, displaying the output in a CLCD window.
- To exit the simulation, press **Ctrl+C**.
- To run `isim_system` in System Canvas:
 - Click Run.
 - 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

- Click OK. The ISIM starts running, displaying the output in a CLCD window.
- 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 Canvas tutorial](#)

[Fast Models glossary](#)

[LISA examples](#) on page 4788

5.4 LISAPlus examples

Source code and System Canvas project files for some example LISA+ components.

Table 5-4: 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 158

[PVBusMaster](#) on page 159

[PVBusSlave](#) on page 162

5.5 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 5-5: 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 readme 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_ELO when data is written. Updates the TXFull bit in MDSCR to indicate the data was read. See the readme 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 GenericCounter .
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 GenericTrace .
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 Trace Cortex-M software with the Instrumentation Trace Macrocell (ITM) on Arm Community.

Example	Description
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 ListTraceSources .
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](#)

5.6 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 5-6: SystemCExport examples

Directory	Description
Bridges	LISA+ source for bridge components.
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	<p>LISA+ source and makefiles for EVS platform examples.</p> <p>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.</p> <p>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.</p> <p>The Dhrystone images <code>dhrystone_v8.axf</code> and <code>dhrystone.axf</code> for Armv7, are provided in the Fast Models Third Party IP package. They are loaded from <code>\$PVLIB_HOME/images/</code>.</p> <p>For more information about building EVS platforms, see Build and run an EVS platform example.</p>

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 Building an SVP 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. There are three subdirectories:</p> <p>SVP_Base Armv8 Base Platform SVPs.</p> <p>SVP_bigLITTLE Armv7 big.LITTLE™ SVP, containing Cortex®-A15 and Cortex®-A7 clusters.</p> <p>SVP_LinuxBoot Minimal Armv7 platform that is suitable for booting Linux, containing a Cortex®-A15 cluster.</p>

5.7 Build and run an EVS platform example

The SystemCExport examples are located under `$PVLIB_HOME/examples/SystemCExport/`.

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.



These instructions also apply to the SVP platform examples, which are located in `$PVLIB_HOME/examples/SystemCExport/SVP_Platforms`.

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_Base/Build_Cortex-A55
make rel_gcc93_64
```

This command creates the target executable `EVS_Base_Cortex-A55.x` and copies the shared objects that are required to run it into the current directory.

2. To see a full list of command-line options for the EVS platform, run it with the `--help` option:

```
./EVS_Base_Cortex-A55.x --help
```

The following example command-line shows how to load an application on the platform:

```
./EVS_Base_Cortex-A55.x \  
-a Base.cluster0.*=$PVLIB_HOME/images/image.axf \  
-C Base.bp.secure_memory=0
```

where:

-a

is the name of an 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>`.

3. The simulation starts running, displaying the output in a CLCD window.
4. 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.

6. Base Platform

The Base Platform system model allows early development, distribution, and demonstration of software deliverables. A range of Base Platform FVPs are supplied as source code examples in Fast Models and as pre-built executables in the FVP Standard Library package.

For a list of the Base and BaseR Platform FVPs and the instances in them, see [Base Platform FVPs](#) and [BaseR Platform FVPs](#) in the FVP Reference Guide.

See `$PVLIB_HOME/examples/LISA/FVP_Base/` for LISA+ source and project files for the Base Platform FVP examples.

The standard peripheral set enables software development and porting. The platform is an evolution of the VE Fixed Virtual Platforms (FVPs), based on the Arm® Versatile Express (VE) hardware development platform.

It provides:

- Two configurable clusters of up to eight core models that implement:
 - AArch64 at all exception levels.
 - Configurable AArch32 support at all exception levels.
 - Configurable support for little and big endian at all exception levels.
 - Generic timers.
 - Self-hosted debug.
 - CADI debug.
 - GICv3 memory-mapped processor interfaces and distributor.
- Peripherals for multimedia or networking environments.
- Four PL011 UARTs.
- A CoreLink™ CCI-400 Cache Coherent Interconnect, or CCI-550 in Base Platform RevC.
- Architectural GICv3 model.
- High Definition LCD Display Controller, 1920*1080 resolution at 60fps, with single I2S and four stereo channels.
- 64MB NOR flash and board peripherals.
- CoreLink™ TZC-400 TrustZone® Address Space Controller.

Related information

- [LISA examples](#)

6.1 Base Platform RevC

Base Platform RevC is an evolution of the Base Platform, enhanced to support exploration of system-level virtualization.

Base Platform RevC has the following additions to the Base Platform:

- A PCIeRootComplex with these address regions:
 - A PCI-E config region at 0x0040000000 to 0x004ffffffffff. The PCI-E config region implements ECAM.
 - A PCI-E memory region at 0x0050000000 to 0x005ffffffffff.
 - A PCI-E memory region at 0x4000000000 to 0x7ffffffffff.
- An SMMUv3AEM with a control region at 0x002B400000 to 0x002B4FFFFFF.

The SMMUv3 is placed so that accesses to memory by PCI devices acting as bus masters are affected by it.

- A DMA330x4 with a control region at 0x002B500000 to 0x002B5FFFFFF.
- A Corelink™ CCI550 Cache Coherent Interconnect. Base Platform has a CCI-400.
- An AHCI controller model including a SATA disk model.
- An SMMUv3TestEngine component with a control region at 0x002bfe0000 to 0x002bffffffffff. This component is a traffic generator that acts as a (secure) device upstream of the SMMUv3.
- Two PCIe virtio devices are above the SMMU. By default they are configured to be device 0 and 2 on bus 0.
- The PCIe devices use a DeviceID that is the same as their RequestorID (BDF).
- Base Platform RevC AEMvA FVPs include a Mali_G76 GPU.
- Base Platform RevC AEMvA FVPs implement Armv8.0 by default, which does not support the Statistical Profiling Extension (SPE). To include SPE, add parameter `cluster0.has_arm_v8-3=1`, or similar, to the command line.

LISA+ source and project files for Base Platform RevC FVPs are located in `$PVLIB_HOME/examples/LISA/FVP_Base_RevC/`.

The FVP_Base_RevC-2xAEMvA model is freely available and does not require a license from Arm. You can download it from [Fixed Virtual Platforms](#) on Arm Developer.

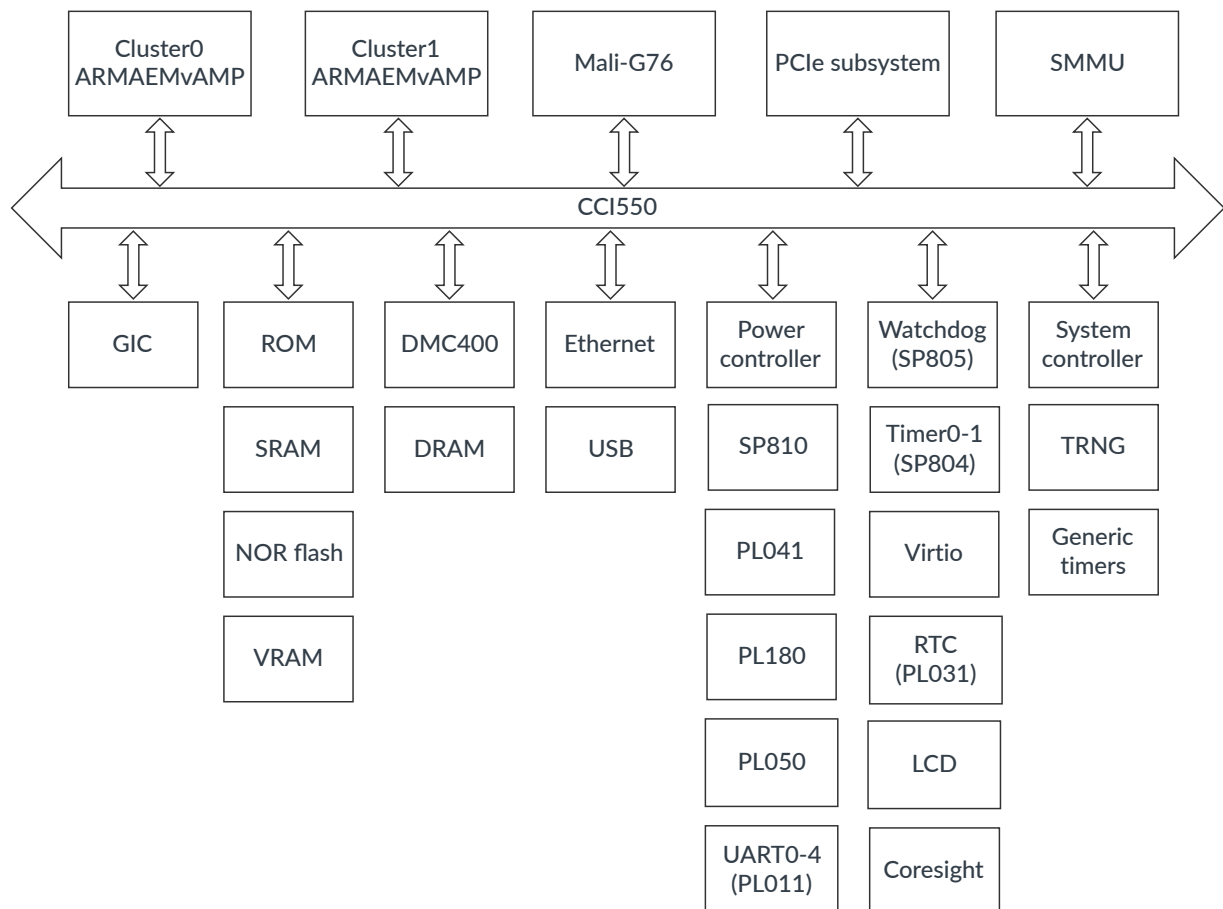
Related information

- [Base Platform memory map](#)
- [Base Platform interrupt assignments](#)

6.1.1 Base Platform RevC block diagram

This diagram shows the main components in Base Platform RevC. The platform shown is FVP_Base_RevC-2xAEMvA.

Figure 6-1: Base Platform RevC block diagram



6.1.2 BasePlatformPCIRevC component

This component is an integrated PCIe subsystem which forms part of the Base Platform RevC. It incorporates an SMMUv3, a PCIe, an AHCI controller, and two PCI devices which wrap a pair of virtio PCI block devices. This model is written in LISA+.



Note

- You can include this component in a platform model, but Arm does not support using its subcomponents to create your own PCIe platform.
- The PCIe is not an implementation of any specific IP, but a functional, and limited, implementation of the PCIe standard.

BasePlatformPCIRRevC is composed of the following model components:

pci.pvbus2pci

The bridge from the Programmer's View bus to the PCI bus.

pci.pcidevice<n>

A wrapper around the underlying virtio block device. There are two block devices in the system, 0 and 1.

pci.pcivirtioblockdevice<n>

The instances of the virtio block device component.

pci.ahci_pci.ahci

An AHCI_SATA component with the following features:

- Each AHCI controller supports up to 32 simulated SATA disks. The configuration parameter `image_path` is a comma-separated list of one or more disk images.
- Supports 64-bit addresses.
- Supports plain, linear disk images, but also works with sparse files.

Some interesting options are:

- If the following options are set to non-zero values, they print messages about the operation of the bridge. The higher the value, the more verbose the component is:

```
pci.pvbus2pci.diagnostics=0x0 # (int) default = '0x0': Diagnostics level: [0x0..0x4]
pci.pcidevice<N>.diagnostics=0x0 # (int) default = '0x0': Diagnostics level:
[0x0..0x4]
```

- Each PCI device uses three BARs; one for config space, one for the MSI-X table structure and one for the MSI-X Pending Bit Array. Each of these can be configured to be 32 bits or 64 bits wide.

The Bus and Device number can be configured for each PCI device. If the device advertises MSI-X, support can be configured.

```
pci.pcidevice<N>.bus=0x0 # (int ) default = '0x0' : Bus number for this device :
[0x0..0xFF]
pci.pcidevice<N>.device=0x0 # (int ) default = '0x0' : Device number on this bus :
[0x0..0x1F]
pci.pcidevice<N>.bar0_64bit=0 # (bool) default = '0' : If BAR 0 is 64 bits wide, if
region size is nonzero
pci.pcidevice<N>.msix_support=0 # (bool) default = '0' : Enable device support for
MSI-X
pci.pcidevice<N>.bar2_64bit=0 # (bool) default = '0' : If BAR 2 is 64 bits wide, if
region size is nonzero
pci.pcidevice<N>.bar4_64bit=0 # (bool) default = '0' : If BAR 4 is 64 bits wide, if
region size is nonzero
```

- The following option configures the image file that the virtio block device exposes:

```
pci.pcivirtioblockdevice<N>.image_path="" # (string) default = '' : image file path
```

- There are two `PVBusLogger`s in the `pdbus2pci` component. One is in front of the Configuration space and one is in front of the Device space:

```
FVP_Base_AEMvA_AEMvA-PCI.pci.pvbus2pci.devicellogger
FVP_Base_AEMvA_AEMvA-PCI.pci.pvbus2pci.cfglogger
```

- There is one `PVBusLogger` in the `pcidevice` component. This reports on DMA accesses by the PCI device:

```
FVP_Base_AEMvA_AEMvA-PCI.pci.pcidevice0.dmalogger
```

- There is a `PVBusLogger` downstream of the SMMU. This reports on the transactions after they have been transformed by the SMMU:

```
FVP_Base_AEMvA_AEMvA-PCI.pci.smmullogger
```

Use the [GenericTrace](#) plug-in to capture traces from the loggers. For example, to see all accesses to device space, add the following options to the command line:

```
--plugin GenericTrace.so
-C TRACE.GenericTrace.trace-sources="FVP_Base_AEMvA_AEMvA-PCI.pci.pvbus2pci.devicellogger.*"
```

Use the [ListTraceSources](#) plug-in to list all the available trace sources in the model.

- To supply the AHCI controller with one or more SATA disk images, use the `image_path` parameter. For example:

```
-C pci.ahci_pci.ahci.image_path=disk1tb.img,disk8tb.img
```

Table 6-1: BasePlatformPCIRevC ports

Name	Protocol	Type	Description
<code>smmu_incoming_pvbus_s</code>	PVBus	Slave	Input port to smmu incoming device traffic slave based on PVBus protocol.
<code>pdbus_address_map_s</code>	PVBus	Slave	Input port to service transactions based on the PVBus protocol.
<code>pdbus_address_map_m</code>	PVBus	Master	Output port to send out PVBus transactions that are not handled by this component.
<code>system_reset</code>	Signal	Slave	Input port to handle reset signals. It is used to reset the internal state of this component.
<code>sev_out</code>	Signal	Peer	Port to send out a notification of the occurrence of an event as <code>sg::Signal</code> to a peer.
<code>interrupts[224]</code>	Signal	Master	Array of output ports of type <code>sg::Signal</code> to send out interrupts generated by this component.
<code>pdbus_pci_dma_m</code>	PVBus	Master	Output port to send out any DMA (of PVBus protocol) accesses originating from this component.
<code>clk_in</code>	ClockSignal	Slave	Input port to connect to a ClockSignal provider.

Table 6-2: BasePlatformPCIRevC parameters

Name	Type	Allowed values	Default value	Description
ITS0-base	uint64_t	0 - 0xFFFFFFFFFFFFFFFF	0x2f020000	The ITS0 Base address.
pci_smmuv3. mmu.SMMU_IDR1	uint32_t	0 - 0xFFFFFFFF	0xe739d10	SMMU_IDR1.
pci_smmuv3. mmu.smmu_ msi_device_id	uint32_t	0 - 0xFFFFFFFF	0x10000	When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID. See parameters <code>msi_attribute_transform</code> and <code>enable_device_id_checks</code> .

Related information[SMMUv3AEM](#) on page 4597[VirtioBlockDevice](#) on page 3291

6.1.3 Base Platform RevC PCIeRootComplex

This component is an abstraction that represents the root of a PCIe device tree.

The addressable regions are also available as the following model parameters:

- PCIE_CFG_START, PCIE_CFG_END
- PCIE_MEM0_START, PCIE_MEM0_END
- PCIE_MEM1_START, PCIE_MEM1_END

PCIe device types:

Bridge

For example Root Port, Switch.

Endpoint

For example AHCI_SATA, SMMUv3TestEngine.

The PCIe device tree has the following properties:

- It always has PCIeRootComplex (RC) as its root.
- RC can connect to a Bridge or to an EndPoint.

PCIe device tree specification:

- The entire PCIe device tree can be specified through the model parameter `hierarchy_file_name` whose format is JSON. If no file is supplied, a default tree is assumed.

- Start the model and use the MTI trace source `ROOT_COMPLEX_HIERARCHY` to see the exact tree and parameter used for each device in the default tree. For example:

```
-C TRACE.GenericTrace.trace-
sources="FVP_Base_RevC_2xAEMvA.pci.pcie_rc.ROOT_COMPLEX_HIERARCHY"
```

- The top-level devices of the default tree are: `AHCI_SATA`, `HostBridge`, `ROOTPORT0`, `ROOTPORT1`, `ROOTPORT2`, `ROOTPORT3`, `SMMUv3TestEngine0`, `SMMUv3TestEngine1`. `ROOTPORT1-4` are of type `PCIEBridge` with parameter `port_type = 4`.

Error responses from PCIe devices:

- AMBA SLVERR (TX_ABORT) to PCIe Completer Abort (CA)
- AMBA DECERR (TX_DECODEABORT) to PCIe Unsupported Request (UR)

6.1.4 Base Platform RevC SMMUv3AEM

This is an architectural model implementing the SMMUv3.0 and SMMUv3.1 architectures which are for I/O virtualization of devices. The SMMU is placed so that accesses to memory by PCI devices acting as bus masters are affected by it.

The SMMU has the following features:

- Memory that is mapped to the range `0x2B400000-0x2B4FFFFFF`.
- Interrupts with IRQ IDs in the range 103-111.
- The event output pin of the SMMU is passed to the clusters.
- The downstream ports of the SMMU attach to the coherent bus infrastructure and so are coherent with the core clusters. All cores and the SMMU are in the same shareability domain. There is no distinction between the inner and outer shareability domains.
- The parameters of the SMMU determine its capabilities and have default values which can be overwritten if necessary.
- The SMMU is configured to only accept 16-bit StreamIDs and there is a 1:1 correspondence between RequestorID and StreamID.
- By default, the SMMU uses DeviceID `0x10000` to identify itself to the GIC (`pci.pci_smmuv3.mmu.smmu_msi_device_id`).

The parameter `gic_distributor.ITS-device-bits` is set to 17 by default to support the 17-bit DeviceIDs.

The SMMU has the following limitations:

- It does not support RAS.
- 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.

6.1.5 Base Platform RevC PCIe device assignment

Every PCIe Endpoint can be enabled to support this feature which makes the Endpoint assignable to a Trusted Compute Base (TCB).

When an Endpoint is assigned to a TCB, it can be configured to both handle and generate memory transactions to specific memory regions.

The model has the following features:

- DOE read/write mailbox.
- SPDMM messages including FINISH_RESPONSE. There is no encryption of messages yet.
- IDE capability on device and root-port.
- Secure-SPDM. There is no MAC, encryption.
- TDISP messages and state machine.

To enable the device assignment feature in a PCIe RootPort, use the `rootportda_supported` parameter. To enable the feature in a PCIe Endpoint, use the following parameters:

doe_supported

Enables the PCIe extended capability called Data Object Exchange.

ide_supported

Enables the PCIe extended capability called Integrity and Data Encryption. This parameter also exists on the PCIe RootPort. The FVP does not model any encryption details but responds appropriately for “IDE” request messages.

x509_cert_der_filename

Filename containing an X-509 certificate issued for this DOE device in DER format. This certificate should contain the corresponding public key of the pair containing parameter `<rsa_priv_key_pem_filename>`.

rsa_priv_key_pem_filename

Filename containing a RSA private key, in PEM, for this DOE device. The corresponding RSA public key should be used to create parameter `<x509_cert_der_filename>`.

dmtf_meas_spec_info

Information about the DMTF measurement specification format. See the DMTF measurement specification format table for more information. Use the following format for the content of each line in this file:

`index_num_in_dec,dmtf_meas_spec_value_type,measurement_filename`

For example:

```
0,8,meas1.txt
1,9,meas2.txt
10,0,meas3.txt
```


6.1.6 Base Platform RevC legacy PCI interrupts

Each PCI device is hardwired to use INTA, with a value of 1 in the `interrupt_pin` register. This is required by the PCI specification for single-function devices.

The interrupts in the PCI host bridge are mapped according to section 2.2.6 of the PCI Local Bus Specification Revision 3.0, using the following formula, where the values for DeviceInterrupt are INTA = 0, INTB = 1, INTC = 2, INTD = 3:

$$\text{BridgeInterrupt} = (\text{Device} + \text{DeviceInterrupt}) \% 4$$

This formula produces the following mappings:

Table 6-3: Mapping BridgeInterrupts to DeviceInterrupts

BridgeInterrupt ID	DeviceInterrupt
200	INTA
201	INTB
202	INTC
203	INTD
207	SERR

6.1.7 Base Platform RevC MSI-X

The model optionally implements MSI-X, depending on whether the parameter `msix_support` is set. If set, an MSI-X capability is advertised as a PCI capability.



The Virtio specification is not fully compliant with the PCI specification and the virtio block device cannot be used in a 'pure polling' mode where MSI-X is always masked and only polling the Pending Bit Array is used.

The MSIs produced by the models, when directed to the GIC, have their payload rewritten to carry the DeviceID of the originating device to the GIC.

6.2 Base Platform startup configuration for v9 cores

Use the parameters `pctl.use_in_cluster_ppu` and `cluster.core_power_on_by_default` to configure startup of Arm®v9 cores in the Base Platform.

The following table shows the effect of each combination of these parameters.

Table 6-4: Configuring startup behavior for v9 cores in the Base Platform

Parameter configuration	Base Platform behavior	Recommended?
<code>pctl.use_in_cluster_ppu=true</code> <code>cluster.core_power_on_by_default=true</code>	All cores start up regardless of the PPU configuration done by the power controller. This configuration is invalid.	No
<code>pctl.use_in_cluster_ppu=false</code> <code>cluster.core_power_on_by_default=true</code>	Power controller loses the ability to do power state transitions altogether. All cores start running at once.	No
<code>pctl.use_in_cluster_ppu=true</code> <code>cluster.core_power_on_by_default=false</code>	<p>Only cores that are mentioned in the startup quad parameter are brought up.</p> <p>This configuration has the limitation that the RVBAR address must be supplied as a parameter on the command line.</p> <p>The application start address initialized by the application loader is overridden by the reset vector address provided in the parameter.</p>	Yes, but note the limitation described
<code>pctl.use_in_cluster_ppu=false</code> <code>cluster.core_power_on_by_default=false</code>	Wrong configuration. No core starts up.	No

6.3 Base Platform memory

This section describes the memory of the Base Platform.

6.3.1 Base Platform memory map

The Base Platform memory map is based on the Versatile™ Express RS2 memory map with extensions.



For an explanation of the values in the Security column, see [Base Platform secure memory](#).

Table 6-5: Base Platform memory map

Peripheral	Start address	Size	End address	Security
Trusted Boot ROM, secure flash, IntelStrataFlashJ3	0x00_0000_0000	64MB	0x00_03FF_FFFF	S
Trusted SRAM. 512KB is the default size. To set SRAM to 256KB instead, set the parameter <code>bp.secure_sram_size</code> to 0.	0x00_0400_0000	512KB	0x00_0407_FFFF	S
Trusted DRAM	0x00_0600_0000	32MB	0x00_07FF_FFFF	S

Peripheral	Start address	Size	End address	Security
NOR flash, flash0, IntelStrataFlashJ3	0x00_0800_0000	64MB	0x00_0BFF_FFFF	S/NS
NOR flash, flash1, IntelStrataFlashJ3	0x00_0C00_0000	64MB	0x00_0FFF_FFFF	S/NS
CS1-Pseudostatic RAM, PSRAM, on the motherboard.	0x00_1400_0000	64MB	0x00_17FF_FFFF	S/NS
VRAM	0x00_1800_0000	32MB	0x00_19FF_FFFF	S/NS
Ethernet, SMSC 91C111	0x00_1A00_0000	16MB	0x00_1AFF_FFFF	S/NS
USB, unimplemented	0x00_1B00_0000	16MB	0x00_1BFF_FFFF	S/NS
VE System Registers	0x00_1C01_0000	64KB	0x00_1C01_FFFF	S/NS
System Controller, SP810	0x00_1C02_0000	64KB	0x00_1C02_FFFF	S/NS
AACI, PL041	0x00_1C04_0000	64KB	0x00_1C04_FFFF	S/NS
MCI, PL180	0x00_1C05_0000	64KB	0x00_1C05_FFFF	S/NS
KMI - Keyboard, PL050	0x00_1C06_0000	64KB	0x00_1C06_FFFF	S/NS
KMI - Mouse, PL050	0x00_1C07_0000	64KB	0x00_1C07_FFFF	S/NS
UART0, PL011	0x00_1C09_0000	64KB	0x00_1C09_FFFF	S/NS
UART1, PL011	0x00_1C0A_0000	64KB	0x00_1C0A_FFFF	S/NS
UART2, PL011	0x00_1C0B_0000	64KB	0x00_1C0B_FFFF	S/NS
UART3, PL011	0x00_1C0C_0000	64KB	0x00_1C0C_FFFF	S/NS
Watchdog, SP805	0x00_1C0F_0000	64KB	0x00_1C0F_FFFF	S/NS
Base Platform Power Controller	0x00_1C10_0000	64KB	0x00_1C10_FFFF	S/NS
Dual-Timer 0, SP804	0x00_1C11_0000	64KB	0x00_1C11_FFFF	S/NS
Dual-Timer 1, SP804	0x00_1C12_0000	64KB	0x00_1C12_FFFF	S/NS
Virtio block device	0x00_1C13_0000	64KB	0x00_1C13_FFFF	S/NS
Virtio Plan 9 device	0x00_1C14_0000	64KB	0x00_1C14_FFFF	S/NS
Virtio net device	0x00_1C15_0000	64KB	0x00_1C15_FFFF	S/NS
Real-time Clock, PL031	0x00_1C17_0000	64KB	0x00_1C17_FFFF	S/NS
CF Card, unimplemented	0x00_1C1A_0000	64KB	0x00_1C1A_FFFF	S/NS
Color LCD Controller, PL111	0x00_1C1F_0000	64KB	0x00_1C1F_FFFF	S/NS
VirtioRNG entropy device	0x00_1C20_0000	64KB	0x00_1C20_FFFF	S/NS
LS64TestingFIFO	0x00_1D00_0000	64KB	0x00_1D00_FFFF	S/NS
Utility bus for DSU-110-enabled platforms	0x00_1E00_0000	16MB	0x00_1EFF_FFFF	NS
Non-trusted ROM, nontrustedrom	0x00_1F00_0000	4KB	0x00_1F00_0FFF	S/NS
CoreSight™ and peripherals	0x00_2000_0000	128MB	0x00_27FF_FFFF	S/NS
REFCLK CNTControl, Generic Timer	0x00_2A43_0000	4KB	0x00_2A43_0FFF	S
EL2 Generic Watchdog Control	0x00_2A44_0000	4KB	0x00_2A44_0FFF	S/NS
EL2 Generic Watchdog Refresh	0x00_2A45_0000	4KB	0x00_2A45_0FFF	S/NS
Trusted Watchdog, SP805	0x00_2A49_0000	64KB	0x00_2A49_FFFF	S
TrustZone® Address Space Controller, TZC-400	0x00_2A4A_0000	4KB	0x00_2A4A_0FFF	S
REFCLK CNTRead, Generic Timer	0x00_2A80_0000	4KB	0x00_2A80_0FFF	S/NS
AP_REFCLK CNTCTL, Generic Timer	0x00_2A81_0000	4KB	0x00_2A81_0FFF	S/NS
AP_REFCLK CNTBase0, Generic Timer	0x00_2A82_0000	4KB	0x00_2A82_0FFF	S
AP_REFCLK CNTBase1, Generic Timer	0x00_2A83_0000	4KB	0x00_2A83_0FFF	S/NS

Peripheral	Start address	Size	End address	Security
DMC-400 CFG, unimplemented	0x00_2B0A_0000	64KB	0x00_2B0A_FFFF	S/NS
SMMUv3 AEM (Base Platform RevC only)	0x00_2B40_0000	1MB	0x00_2B4F_FFFF	S/NS
DMA330x4 (Base Platform RevC only).	0x00_2B50_0000	1MB	0x00_2B5F_FFFF	S/NS
GIC Physical CPU interface, GICC. To configure the address of this region use model parameters. See [GICv3IRI].	0x00_2C00_0000	8KB	0x00_2C00_1FFF	S/NS
GIC Virtual Interface Control, GICH. To configure the address of this region use model parameters. See [GICv3IRI].	0x00_2C01_0000	8KB	0x00_2C01_1FFF	S/NS
GIC Virtual CPU Interface, GICV. To configure the address of this region use model parameters. See [GICv3IRI].	0x00_2C02_F000	8KB	0x00_2C03_0FFF	S/NS
CCI-400	0x00_2C09_0000	64KB	0x00_2C09_FFFF	S/NS
Mali™-G76 (Base Platform RevC only)	0x00_2D00_0000	16MB	0x00_2DFF_0000	S/NS
Non-trusted SRAM	0x00_2E00_0000	64KB	0x00_2E00_FFFF	S/NS
GICv3 IRI GICD. To configure the address of this region use model parameters. See [GICv3IRI].	0x00_2F00_0000	64KB	0x00_2F00_FFFF	S/NS
GICv3 IRI GITS. To configure the address of this region use model parameters. See [GICv3IRI].	0x00_2F02_0000	128KB	0x00_2F03_FFFF	S/NS
GICv3 IRI GICR. To configure the address of this region use model parameters. See [GICv3IRI].	0x00_2F10_0000	1MB	0x00_2F1F_FFFF	S/NS
PCIe config region (Base Platform RevC only)	0x00_4000_0000	256MB	0x00_4FFF_FFFF	S/NS
PCIe memory region 1 (Base Platform RevC only)	0x00_5000_0000	256MB	0x00_5FFF_FFFF	S/NS
Trusted Random Number Generator	0x00_7FE6_0000	4KB	0x00_7FE6_0FFF	S
Trusted Non-volatile counters	0x00_7FE7_0000	4KB	0x00_7FE7_0FFF	S
Trusted Root-Key Storage	0x00_7FE8_0000	4KB	0x00_7FE8_0FFF	S
DDR3 PHY, unimplemented	0x00_7FEF_0000	64KB	0x00_7FEF_FFFF	S/NS
HD LCD Controller, PL370	0x00_7FF6_0000	4KB	0x00_7FF6_0FFF	S/NS
DRAM. Memory Tagging Extension (MTE) is supported.	0x00_8000_0000	2GB	0x00_FFFF_FFFF	P
DRAM. Memory Tagging Extension (MTE) is supported.	0x08_8000_0000	30GB	0x0F_FFFF_FFFF	P
PCIe memory region 2 (Base Platform RevC only)	0x40_0000_0000	256GB	0x7F_FFFF_FFFF	S/NS
DRAM. Memory Tagging Extension (MTE) is supported.	0x88_0000_0000	480GB	0xFF_FFFF_FFFF	P
DRAM. Memory Tagging Extension (MTE) is supported.	0x00_0880_0000_0000	7.5TB	0x00_0FFF_FFFF_FFFF	P
DRAM. Memory Tagging Extension (MTE) is supported.	0x00_8800_0000_0000	120TB	0x00_FFFF_FFFF_FFFF	P
DRAM. Memory Tagging Extension (MTE) is supported.	0x08_8000_0000_0000	1920TB	0x0F_FFFF_FFFF_FFFF	P
DRAM. Memory Tagging Extension (MTE) is supported.	0x88_0000_0000_0000	2PB	0x8F_FFFF_FFFF_FFFF	P

Related information

[BaseR Platform memory map](#) on page 4809

6.3.2 Base Platform secure memory

Use the `bp.secure_memory` parameter to enable security checking on memory transactions by the TZC-400.

Table 6-6: Secure and Non-secure access permissions

Security	<code>bp.secure_memory=false</code>	<code>bp.secure_memory=true</code>
S	Secure and Non-secure access permitted.	Secure access is permitted, Non-secure access aborts.
S/NS	Secure and Non-secure access permitted.	Secure and Non-secure access permitted.
P	Secure and Non-secure access permitted.	Access conditions are programmable by the TZC-400.



Note

- The default state of the TZC-400 is to abort all accesses, even from Secure state.
- Setting both `bp.secure_memory` and `bp.has_rme` parameters to 1 is invalid and produces a warning message.

Table 6-7: NSAIDs and filters that masters present to the TZC-400

Component	Non-Secure Access Identity (NSAID)	TZC-400 filter unit number (0-3)
Cluster 0	9	0
Cluster 1	9	0
VirtioNetMMIO	9	0
VirtioP9Device	8	0
VirtioBlockDevice	8	0
PL111_CLCD	7	2
HDLCD0	2	2

6.3.3 BaseR Platform memory map

The BaseR Platform copies its memory map from the Base Platform, but swaps the upper 2GB of address space with the lower 2GB.

Therefore:

- Any peripherals in the memory range `0x0` to `0x7FFFFFFF` in the Base Platform are available at the same offset in the memory range `0x80000000` to `0xFFFFFFFF` in the BaseR Platform.
- Memory in the range `0x80000000` to `0xFFFFFFFF` in the Base Platform is available at the same offset in the range `0x0` to `0x7FFFFFFF` in the BaseR Platform. For example, DRAM in the Base Platform memory map starts at address `0x80000000`. In the BaseR Platform, this is mapped to 0.

The reason for this difference is that in the Arm®v8-R architecture, the upper 2GB of memory does not have execution permissions by default. So, code could not run from DRAM after reset if DRAM started at address `0x80000000`.

If the TCMs are enabled, for example with the parameter `-c cluster0.cpu0.tcm.a.enable=1`, then they are located at address 0. To move TCMs away from 0, use the parameters `itcm_base` and `dtcm_base`. For example:

```
-C cluster0.cpu0.itcm_base=0x8000 -C cluster0.cpu0.dtcn_base=0x18000
```

Related information

[Base Platform memory map](#) on page 4806

6.3.4 Base Platform DRAM

The multiple DRAM regions do not alias each other and form a contiguous 4PB area. The total amount of DRAM on the Base Platform system model is configurable. This ability affects where usable DRAM appears.

If the Base Platform system model has `bp.dram_size=4`, the default, then 2GB of DRAM is accessible at `0x00_8000_0000` to `0x00_FFFF_FFFF`, and the remaining 2GB is accessible at `0x08_8000_0000` to `0x08_FFFF_FFFF`.

If, instead, the Base Platform system model has `bp.dram_size=8`, then 2GB of DRAM is accessible at `0x00_8000_0000` to `0x00_FFFF_FFFF` and the remaining 6GB is accessible at `0x08_8000_0000` to `0x09_FFFF_FFFF`.

The default contents of RAM not otherwise written by the simulation is a repeating sequence of the following 64-bit value: `CFDFDFDFDFDFDFCF`.



Memory is allocated on demand, and performance degrades if very large amounts of memory are used.

6.4 Base Platform interrupt assignments

The platform assigns the Shared Peripheral Interrupts (SPIs) and Private Peripheral Interrupts (PPIs) on the GIC.



- SPI and PPI numbers are mapped onto GIC interrupt IDs as described in the Arm Generic Interrupt Controller Specification.
- IRQ IDs 103-111, 192-194, and 200-207 apply to the Base Platform RevC only.

Table 6-8: SPI GIC assignments

IRQ ID	SPI offset	Device
32	0	Watchdog, SP805.
34	2	Dual-Timer 0, SP804.
35	3	Dual-Timer 1, SP804.
36	4	Real-time Clock, PL031.
37	5	UART0, PL011.
38	6	UART1, PL011.
39	7	UART2, PL011.
40	8	UART3, PL011.
41	9	MCI, PL180, MCIINTRO.
42	10	MCI, PL180, MCIINTR1.
43	11	AACI, PL041.
44	12	KMI - Keyboard, PL050.
45	13	KMI - Mouse, PL050.
46	14	Color LCD Controller, PL111.
47	15	Ethernet, SMSC 91C111.
56	24	Trusted Watchdog, SP085.
57	25	AP_REFCLK, Generic Timer, CNTPSIRQ.
58	26	AP_REFCLK, Generic Timer, CNTPSIRQ1.
59	27	EL2 Generic Watchdog WS0.
60	28	EL2 Generic Watchdog WS1.
74	42	Virtio block device.
75	43	Virtio P9 device.
76	44	Virtio net device.
78	46	VirtioRNG entropy device.
80	48	TZC-400.
92	60	cluster0.cpu0 PMUIRQ.
93	61	cluster0.cpu1 PMUIRQ.
94	62	cluster0.cpu2 PMUIRQ.
95	63	cluster0.cpu3 PMUIRQ.
96	64	cluster1.cpu0 PMUIRQ.
97	65	cluster1.cpu1 PMUIRQ.
98	66	cluster1.cpu2 PMUIRQ.
99	67	cluster1.cpu3 PMUIRQ.
103	71	SMMUv3 non-secure combined interrupt. Base Platform RevC only.
104	72	SMMUv3 secure combined interrupt. Unused because there is no secure side. Base Platform RevC only.
105	73	SMMUv3 secure event queue. Unused because there is no secure side. Base Platform RevC only.
106	74	SMMUv3 non-secure event queue. Base Platform RevC only.
107	75	SMMUv3 PRI queue. Unused because no PCIe device supports PRI. Base Platform RevC only.
108	76	SMMUv3 secure command queue sync. Unused because there is no secure side. Base Platform RevC only.

IRQ ID	SPI offset	Device
109	77	SMMUv3 non-secure command queue sync. Base Platform RevC only.
110	78	SMMUv3 secure GERROR. Unused because there is no secure side. Base Platform RevC only.
111	79	SMMUv3 non-secure GERROR. Base Platform RevC only.
117	85	HD LCD Controller, PL370.
139	107	Trusted Random Number Generator.
192	160	Mali™-G76 GPU. Base Platform RevC only.
193	161	Mali™-G76 GPU job. Base Platform RevC only.
194	162	Mali™-G76 GPU MMU. Base Platform RevC only.
200	168	PCIe INTA. Base Platform RevC only.
201	169	PCIe INTB. Base Platform RevC only.
202	170	PCIe INTC. Base Platform RevC only.
203	171	PCIe INTD. Base Platform RevC only.
207	175	PCIe SERR. Base Platform RevC only.

Table 6-9: PPI GIC assignments

IRQ ID	PPI offset	Device
19	3	Secure hypervisor virtual timer interrupt
20	4	Secure hypervisor physical timer interrupt
21	5	Statistical Profiling Unit (SPU) interrupt
22	6	DCC, comms channel, interrupt
23	7	PMU, performance counter, overflow
24	8	Cross Trigger Interface (CTI) interrupt
25	9	Virtual CPU interface maintenance interrupt
26	10	Hypervisor timer interrupt
27	11	Virtual timer interrupt
28	12	Hypervisor virtual timer interrupt
29	13	Secure physical timer interrupt
30	14	Non-secure physical timer interrupt
31	15	Trace Buffer Unit (TRBU) interrupt

6.5 Base Platform clocks

This table shows the clock frequencies of the Base Platform peripherals.

Table 6-10: Peripheral clock frequencies in the Base Platform

Device	Clock
Clusters	100MHz
REFCLK CNTControl, Generic Timer	100MHz
AP_REFCLK CNTCTL, Generic Timer	100MHz
Dual-Timer 0-1, SP804	35MHz

Device	Clock
VE system registers	24MHz
UART 0-3, PL011	24MHz
KMI 0-1, PL050	24MHz
MCI, PL180	24MHz
AACI, PL041	24MHz
Ethernet, SMSC 91C111	24MHz
Watchdog, SP805	24MHz
Color LCD Controller, PL111	23.75MHz
HD LCD Controller, PL370	10MHz
Trusted Watchdog, SP805	32.768kHz
Real-time Clock, PL031	1Hz

6.6 Base Platform parameters

This section describes the parameters.

Table 6-11: Base Platform parameters

Parameter	Type	Allowed values	Default value	Description
<code>bp.dram_size</code>	int	2, 4, or 8-4000000	4	Size of main memory in gigabytes: 2, 4, or any value between 8 and 4000000.
<code>bp.proc_id<n></code> . Some platforms do not expose this parameter.	uint32_t	-	Platform-specific	Processor ID for VE_SysRegs SYS_PROCID<n>.
<code>bp.secure_memory</code>	bool	true, false	true	The security state of the processor limits access to peripherals and RAM. See Note below table.
<code>bp.variant</code> . Some platforms do not expose this parameter.	uint32_t	0x0-0xF	Platform-specific	Board variant for VE_SysRegs SYS_ID.
<code>cache_state_modelled</code>	bool	true, false	true	Enable d-cache and i-cache state for all components.

When loading an image on an EVS, you might see the following message:



Note

```
Warning: Base.cluster0.cpu0: Uncaught exception, thread terminated
In file: gen/scx_scheduler_mapping.cpp:523
In process: Base.thread_p_5 @ 0 s
```

This warning means that the image is attempting to run from DRAM, which is access-controlled by the TZC_400 component. To disable security checking by the TZC_400, specify `-c Base.bp.secure_memory=false` when running the EVS.

Table 6-12: Base Platform debug parameters

Parameter	Type	Allowed values	Default value	Description
dbgen	bool	true, false	true	Debug authentication signal, dbgen.
niden	bool	true, false	true	Debug authentication signal, niden.
spiden	bool	true, false	true	Debug authentication signal, spiden.
spniden	bool	true, false	true	Debug authentication signal, spniden.

6.7 Base Platform components

These component models implement some of the functionality of the Versatile Express (VE) hardware.

A complete model implementation of a Base Platform system model includes both Base Platform-specific components and generic components such as buses and timers. To see a list of all the component instances in the model, run it with the `--list-instances` option.

6.7.1 Base Platform Base_PowerController component

This section describes the Base_PowerController component.

6.7.1.1 Base_PowerController control interface

The Base Platform Power Controller component provides a basic register interface for software to control the power-up and power-down of cores in the cluster.

Identify cores in the system to the Base_PowerController by writing 24 bits in MPIDR format, providing the following levels of affinity:

Bits [23:16]

Affinity level 2.

Bits [15:8]

Affinity level 1.

Bits [7:0]

Affinity level 0.

Examples of affinity usage are `not_applicable/cluster/processor` and `cluster/processor/thread`.

To identify which cores to power up at startup, use parameter `pctl.startup`.

Specify core affinities with a dotted-quad. Wildcards are allowed. The format depends on the architecture:

- In Armv8.1 and earlier, use:

```
-C pctl.startup=0.0.Y.X
```

where X refers to the core number and Y refers to the cluster number. For example 0.0.0.0 refers to cluster0.cpu0 and 0.0.1.1 refers to cluster1.cpu1. Use wildcards to indicate all cores at an affinity level. For example, to turn on all the cores in cluster 0, use 0.0.0.*.

- In Armv8.2 and later, use:

```
-C pctl.startup=0.Z.Y.X
```

where X refers to the thread number, Y refers to the core number, and Z refers to the cluster number.

6.7.1.2 Base_PowerController registers

This section describes the registers.

Register summary

This section describes the power control registers in order of offset from the base memory address.

Table 6-13: Base_PowerController register summary

Offset	Name	Type	Reset	Width	Description
0x00	PPOFFR	RW	0x---	32	Power Control Processor Off Register
0x04	PPONR	RW	0x---	32	Power Control Processor On Register
0x08	PCOFFR	RW	0x---	32	Power Control Cluster Off Register
0x0C	PWKUPR	RW	0x---	32	Power Control Wakeup Register
0x10	PSYSR	RW	0x---	32	Power Control SYS Status Register

PPOFFR

The Power Control Processor Off Register (PPOFFR) characteristics are: purpose, usage constraints, configurations, and attributes.

Purpose

Processor SUSPEND command when PWKUPR and the GIC are programmed appropriately to provide wakeup events from IRQ and FIQ events to that processor.

Usage constraints

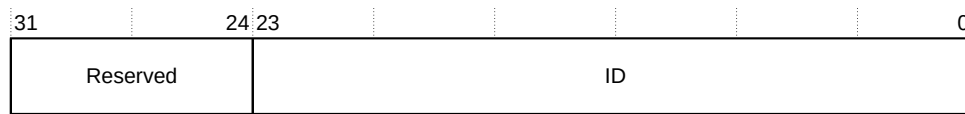
Processor must make power-off requests only for itself.

Configurations

Available in all configurations.

Attributes

See the register summary table.

Figure 6-2: Power Control Processor Off Register bit assignments**Table 6-14: Power Control Processor Off Register bit assignments**

Bits	Name	Function
[31:24]	-	Reserved.
[23:0]	ID	MPIDR format affinity value of the processor to be switched off. Programming error if MPIDR != self.

PPONR

The Power Control Processor On Register (PPONR) characteristics are: purpose, usage constraints, configurations, and attributes.

Purpose

Brings up a processor from low-power mode.

Usage constraints

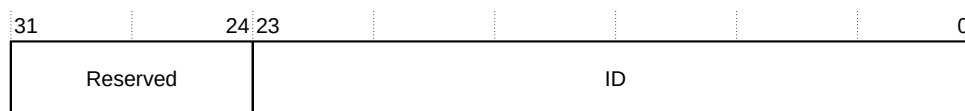
Processor must make power-on requests only for other powered-off processors in the system.

Configurations

Available in all configurations.

Attributes

See the register summary table.

Figure 6-3: Power Control Processor On Register bit assignments**Table 6-15: Power Control Processor On Register bit assignments**

Bits	Name	Function
[31:24]	-	Reserved.
[23:0]	ID	MPIDR format affinity value of the processor to be switched on. Programming error if MPIDR == self.

PCOFFR

The Power Control Cluster Off Register (PCOFFR) characteristics are: purpose, usage constraints, configurations, and attributes.

Purpose

Turns the cluster off.

Usage constraints

Cluster must make power-off requests only for itself.

PSYSR

The Power Control SYS Status Register (PSYSR) characteristics are: purpose, usage constraints, configurations, and attributes.

Purpose

Provides information on the powered status of a given core. Software writes bits [23:0] for the required core and reads the value along with the associated status in bits [31:24].

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

See the register summary table.

Figure 6-6: Power Control SYS Status Register bit assignments

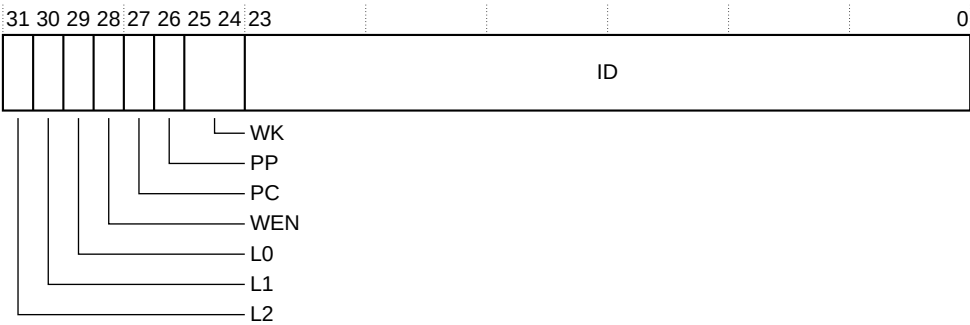


Table 6-18: Power Control SYS Status Register bit assignments

Bits	Name	Function
[31]	L2	Read-only. A value of 1 indicates that affinity level 2 is active/on. If affinity level 2 is not implemented this bit is RAZ .
[30]	L1	Read-only. A value of 1 indicates that affinity level 1 is active/on. If affinity level 1 is not implemented this bit is RAZ .
[29]	L0	Read-only. A value of 1 indicates that affinity level 0 is active/on.
[28]	WEN	Read-only. A value of 1 indicates wakeup interrupts, return from SUSPEND, enabled for this processor. This is an alias of PWKUPR.WEN for this core.
[27]	PC	Read-only. A value of 1 indicates pending cluster off, the cluster enters low-power mode the next time it raises signal STANDBYWFIL2.

Bits	Name	Function
[26]	PP	Read-only. A value of 1 indicates pending processor off, the processor enters low-power mode the next time it raises signal STANDBYWFI.
[25:24]	WK	Read-only. Indicates the reason for LEVEL0 power on: 0b00 Cold power-on. 0b01 System reset pin. 0b10 Wake by PPONR. 0b11 Wake by GIC WakeRequest signal.
[23:0]	ID	MPIDR format affinity value.

6.7.2 Base Platform DebugAccessPort component

This section describes the DebugAccessPort component which models the Debug Access Port (DAP) for external debug connections.

6.7.2.1 DebugAccessPort ports

This section describes the ports.

Table 6-19: DebugAccessPort ports

Name	Protocol	Type	Description
ap_pvbuss_m[2]	PVBus	Master	Debug-access port to bus master channels 0 and 1.
clock	ClockSignal	Slave	Clock input.
paddrdbg31	Signal	Master	Output signal that indicates which master the access came from, AP0 or AP1. Configurable.

6.7.2.2 DebugAccessPort parameters

This section describes the parameters.

Table 6-20: Base Platform DebugAccessPort parameters

Name	Type	Allowed values	Default value	Description
ap0_rom_base_address	uint64_t	0-0xffffffffffffffff	0	ROM base address for AP0.
ap1_rom_base_address	uint64_t	0-0xffffffffffffffff	0	ROM base address for AP1.
ap0_has_debug_rom	bool	true, false	false	AP0 has a Debug ROM.
ap1_has_debug_rom	bool	true, false	false	AP1 has a Debug ROM.

Name	Type	Allowed values	Default value	Description
ap0_set_paddrdbg31	bool	true, false	false	Set paddrdbg31 signal during accesses on AP0.
ap1_set_paddrdbg31	bool	true, false	false	Set paddrdbg31 signal during accesses on AP1.

6.7.3 Simulator visualization - parameters

This section describes the parameters.

Table 6-21: Simulator visualization parameters

Parameter	Allowed values	Default value	Description
cluster0_name	-	"Cluster0"	Cluster0 name.
cluster1_name	-	"Cluster1"	Cluster1 name.
cluster2_name	-	"Cluster2"	Cluster2 name.
cluster3_name	-	"Cluster3"	Cluster3 name.
cpu_name	-	""	Processor name displayed in window title.
diagnostics	true, false	false	Print diagnostic messages.
disable_visualisation	true, false	false	Enable or disable visualization.
rate_limit-enable	true, false	false	Restrict simulation speed so that simulation time more closely matches real time rather than running as fast as possible.
recorder. checkInstructionCount	true, false	true	Checks instruction count in recording file against actual instruction count during playback.
recorder. playbackFileName	-	""	Playback filename. An empty string disables playback.
recorder. recordingFileName	-	""	Recording filename. An empty string disables recording.
recorder. recordingTimeBase	-	0x5F5E100	Timebase in 1/s relative to the master clock. Used for recording. For example, 100 000 000 means 10 nanoseconds resolution simulated time for a 1Hz master clock. Higher values give a higher time resolution. Playback time base is always taken from the playback file.
recorder.verbose	-	0	Enables verbose messages. 0, no messages. 1, normal. 2, more detail.
trap_key	Valid ATKeyCode key value. See \$PVLIB_HOME/include/components/KeyCode.h for a list of values.	0x4a	Trap key that works with left Ctrl to toggle mouse display. The default is the left Alt key, so pressing Left Alt and Left Ctrl simultaneously toggles the mouse display.
window_title	-	"Fast Models - CLCD %cpu%"	Window title. cpu_name replaces %cpu%.

6.7.4 VE_SysRegs component

This section describes the VE system registers component.

6.7.4.1 VE_SysRegs parameters

This table describes the VE_SysRegs parameters.

Table 6-22: VE_SysRegs parameters

Name	Type	Default value	Description
exit_on_shutdown	bool	false	Used to shut down the system. See the Note after the table.
mmbSiteDefault	int	1	Default MultiMedia Bus (MMB) source (0=motherboard, 1=daughterboard 1, 2=daughterboard 2).
sys_proc_id0	int	0x0c000000	Processor ID register at CoreTile Express Site 1.
sys_proc_id1	int	0xff000000	Processor ID at CoreTile Express Site 2.
tilePresent	bool	true	Tile fitted.
user_switches_value	int	0	User switches.



When `exit_on_shutdown` is `true`, if software uses the `sys_CFGCTRL` function `sys_CFG_SHUTDOWN`, then the simulator shuts down and exits. For more information on the `sys_CFGCTRL` function values, see the Motherboard Express µATX V2M-P1 Technical Reference Manual.

6.7.4.2 VE_SysRegs registers

This table describes the configuration registers.

Table 6-23: VE_SysRegs registers

Name	Offset	Access	Description
SYS_ID	0x00	Read/write	System identity
SYS_SW	0x04	Read/write	Bits[7:0] map to switch S6
SYS_LED	0x08	Read/write	Bits[7:0] map to user LEDs
SYS_100HZ	0x24	Read only	100Hz counter
SYS_FLAGS	0x30	Read/write	General purpose flags
SYS_FLAGSCLR	0x34	Write only	Clear bits in general purpose flags
SYS_NVFLAGS	0x38	Read/write	General purpose non-volatile flags
SYS_NVFLAGSCLR	0x3C	Write only	Clear bits in general purpose non-volatile flags
SYS_MCI	0x48	Read only	MCI
SYS_FLASH	0x4C	Read/write	Flash control
SYS_CFGSW	0x58	Read/write	Boot select switch
SYS_24MHZ	0x5C	Read only	24MHz counter

Name	Offset	Access	Description
SYS_MISC	0x60	Read/write	Miscellaneous control flags
SYS_DMA	0x64	Read/write	DMA peripheral map
SYS_PROCID0	0x84	Read/write	Processor ID
SYS_PROCID1	0x88	Read/write	Processor ID
SYS_CFGDATA	0xA0	Read/write	Data to be read/written from/to motherboard controller
SYS_CFGCTRL	0xA4	Read/write	Control data transfer to motherboard controller
SYS_CFGSTAT	0xA8	Read/write	Status of data transfer to motherboard

6.8 Base Platform VE compatibility

Arm expects software that ran on the previous VE model to be compatible with this system model, but you might need to apply some configuration options.

6.8.1 Base Platform GICv2

This system model uses GICv3 by default. You can configure it to support GICv2 or GICv2m.

To configure the model as GICv2m, set the following parameters:

```
-C gicv3.gicv2-only=1 \
-C cluster0.gic.GICD-offset=0x10000 \
-C cluster0.gic.GICC-offset=0x2F000 \
-C cluster0.gic.GICH-offset=0x4F000 \
-C cluster0.gic.GICH-other-CPU-offset=0x50000 \
-C cluster0.gic.GICV-offset=0x6F000 \
-C cluster0.gic.PERIPH-size=0x80000 \
-C cluster1.gic.GICD-offset=0x10000 \
-C cluster1.gic.GICC-offset=0x2F000 \
-C cluster1.gic.GICH-offset=0x4F000 \
-C cluster1.gic.GICH-other-CPU-offset=0x50000 \
-C cluster1.gic.GICV-offset=0x6F000 \
-C cluster1.gic.PERIPH-size=0x80000 \
-C gic_distributor.GICD-alias=0x2c010000
```

To configure the model as GICv2, set the following parameters:

```
-C gicv3.gicv2-only=1 \
-C cluster0.gic.GICD-offset=0x1000 \
-C cluster0.gic.GICC-offset=0x2000 \
-C cluster0.gic.GICH-offset=0x4000 \
-C cluster0.gic.GICH-other-CPU-offset=0x5000 \
-C cluster0.gic.GICV-offset=0x6000 \
-C cluster0.gic.PERIPH-size=0x8000 \
-C cluster1.gic.GICD-offset=0x1000 \
-C cluster1.gic.GICC-offset=0x2000 \
-C cluster1.gic.GICH-offset=0x4000 \
-C cluster1.gic.GICH-other-CPU-offset=0x5000 \
-C cluster1.gic.GICV-offset=0x6000 \
-C cluster1.gic.PERIPH-size=0x8000 \
-C gic_distributor.GICD-alias=0x2c010000
```

To configure MSI frames for GICv2m, parameters are available to set the base address and configuration of each of 16 possible frames. Eight frames are Secure and eight frames are Non-secure:

```
-C gic_distributor.MSI_S-frame0-base=ADDRESS \
-C gic_distributor.MSI_S-frame0-min-SPI=NUM \
-C gic_distributor.MSI_S-frame0-max-SPI=NUM
```

In this example, you can replace `msi_s` with `MSI_NS`, for NS frames, and you can replace `frame0` with `frame1` to `frame7` for each of the possible 16 frames. If the base address is not specified for a given frame, or the SPI numbers are out of range, the corresponding frame is not instantiated.

6.8.2 Base Platform GICv3

If a Base Platform includes an implementation of the GICv3 system registers, it is enabled by default.

The GIC distributor and CPU (core) interface have parameters that allow configuration of the model to match different implementation options. Use `--list-params` to get a full list. Configuration options for the GIC model must be available under:

- `cluster[0-n].gic.*`
- `cluster[0-n].gicv3.*`
- `gic_distributor.*`

6.8.3 Base Platform system global counter

The Generic Timer registers of the cores do not operate by default.

The model provides a memory-mapped interface to the system global counter, and enables the free-running timer from reset. However, the architectural requirement is that such a counter is not enabled at reset. As a result, the Generic Timer registers of the cores do not operate unless either:

- Software enables the counter peripheral by writing the `FCREQ[0]` and `EN` bits in `CNTCR` at `0x2a430000`. Arm recommends this approach.
- The `-c bp.refcounter.non_arch_start_at_default=1` parameter is set. This approach provides compatibility with older software.

6.8.4 Disable security in the Base Platform

Base Platform FVPs have an enhanced security map for peripherals. By default, it restricts access to some peripherals.

Software must program the TZC-400 to make any accesses to DRAM, because the reset configuration blocks all accesses.

For backward compatibility with software that cannot program the TZC-400, this parameter setting permits all accesses regardless of security state:

```
-C bp.secure_memory=false
```

Related information

[Base Platform memory map](#) on page 4806

6.9 Base Platform unsupported VE features

This system model does not support software that relies on some features of the VE model.

6.9.1 Base Platform memory aliasing at 0x08_00000000

The VE model permits an alias of the 2GB region of DRAM between addresses 0x80000000 and 0xFFFFFFFF with addresses 0x08_00000000 to 0x08_7FFFFFFF. The Base Platform does not have this alias and the region 0x08_00000000 to 0x08_7FFFFFFF is Reserved.

6.9.2 Boot ROM alias at 0x00_0800_0000 in the Base Platform

In the VE model, the region at 0x00_0800_0000 was an alias of the trusted boot ROM at 0x00_0000_0000. It is now an independent region of NOR flash.

6.9.3 Changed parameters in the Base Platform

Most parameter names in the VE model have been simplified in the Base Platform.

Components that were previously in motherboard or daughterboard groups are now in a `bp` group. The model does not recognize the previous parameter names.

In a change to the previous default, the Base Platform models the core cache state by default. You can disable this using a single parameter for all cores in the simulation, using the `cache_state_modelled` parameter:

```
-C cache_state_modelled=0
```



Cortex® Base Platforms do not model the cache state by default.

7. Microcontroller Prototyping System 2

This chapter describes the MPS2 system model.

7.1 MPS2

The Microcontroller Prototyping System 2 (MPS2) Fixed Virtual Platforms (FVPs) implement a subset of the functionality of the Arm Versatile Express V2M-MPS2 and V2M-MPS2+ motherboard hardware.

For a list of the MPS2 Platform FVPs and the instances in them, see [MPS2 Platform FVPs](#) in the FVP Reference Guide.

LISA+ source and project files for some MPS2 FVP examples are available in the `$FVLIB_HOME/examples/LISA/FVP_MPS2/` directory.

MPS2 platforms include MPS2-specific components and generic components, such as buses and timers. They are sufficiently accurate to boot the Keil RTX RTOS and run the Blinky application.

To list the model parameters and their types, allowed values, default values, and descriptions, run the model with the `--list-params` argument.

Related information

[AN400 - ARM Cortex-M7 SMM on V2M-MPS2](#)

7.2 MPS2 platform types

Configure the MPS2 FVP platform type using the `fvf_mps2.platform_type` parameter.

It has the following possible values:

0

This value is the default. The FVP acts as a V2M-MPS2 system, with the additions for v8-M, as specified in the Arm®v8-M MPS2 System Specification (ECM 0468897), v0.8. This specification is confidential and is available only to licensed Arm customers. For details, contact your Arm support representative.

1

The FVP acts as an IoT Kit on an MPS2+ board. For details, see the following documents:

- [Cortex®-M23 processor Arm®v8-M IoT Kit User Guide \(ECM 0635473\)](#).
- [Cortex®-M33 processor Arm®v8-M IoT Kit User Guide \(ECM 0601256\)](#).

2

The FVP acts as an Arm® CoreLink™ SSE-200 Subsystem on an MPS2+ board. For details, see [AN521 - Example SSE-200 Subsystem for MPS2+ Application Note](#).

7.3 MPS2 memory maps

This section describes the MPS2 memory maps.

7.3.1 MPS2 memory map for models without the Armv8-M additions

This section describes the MPS2 memory map for older cores, without the Armv8-M additions.

For standard Arm peripherals, see the TRM for that device.



- A bus error is generated for accesses to memory areas not shown in this table.
- Any memory device that does not occupy the total region is aliased within that region.

Table 7-1: Memory map for models without the Armv8-M additions

Description	Modeled	Address range
Ethernet. Through ahb_to_extmem16. Offset 0x0 to 0x0FE for CSRs, 0x100 to 0x1FE for FIFO.	Partial	0xA0000000 to 0xA00FFFFF
PSRAM (16MB)	Yes	0x60000000 to 0x60FFFFFF
VGA Image (512x128) (AHB)	Yes	0x41100000 to 0x4110FFFF
VGA Console (AHB)	Yes	0x41000000 to 0x4100FFFF
Block RAM (boot time). Reserved 64KB, 16K implemented. This memory is wrapped through the region.	Yes	0x40200000 to 0x402FFFFF
Reserved	N/A	0x40030000 to 0x401FFFFF
SCC register	Yes	0x4002F000 to 0x4002FFFF
Reserved	N/A	0x40029000 to 0x4002EFFF
FPGA System Control & I/O, APB	Yes	0x40028000 to 0x40028FFF
Reserved	N/A	0x40025000 to 0x40027FFF
Audio I2S, APB	Partial	0x40024000 to 0x40024FFF
SBCon (Audio Configuration), APB	Yes	0x40023000 to 0x40023FFF
SBCon (Touch for LCD module), APB	Partial	0x40022000 to 0x40022FFF
PL022 (SPI for LCD module), APB	Partial	0x40021000 to 0x40021FFF

Description	Modeled	Address range
PL022 (SPI), APB	Yes	0x40020000 to 0x40020FFF
CMSDK system controller	Yes	0x4001F000 to 0x4001FFFF
Reserved for extra GPIO & other AHB peripherals	N/A	0x40012000 to 0x4001EFFF
CMSDK AHB GPIO #1	Yes	0x40011000 to 0x40011FFF
CMSDK AHB GPIO #0	Yes	0x40010000 to 0x40010FFF
CMSDK APB subsystem	Yes	0x40000000 to 0x4000FFFF
Reserved	N/A	0x20800000 to 0x20FFFFFF
ZBTSRAM 2 & 3 (2x32-bit). Reserved 8MB, 4MB available. The two SRAM blocks are interleaved.	Yes	0x20000000 to 0x207FFFFF
Reserved	N/A	0x01010000 to 0x1FFFFFFF
Reserved	N/A	0x00800000 to 0x00FFFFFF
ZBTSRAM 1 (64-bit). Wrapped. Only 4MB ZBTSRAM fitted.	Yes	0x00400000 to 0x007FFFFF
ZBTSRAM 1 (64-bit)	Yes	0x00004000 to 0x003FFFFF
Mappable memory. When <code>zbt_boot_ctrl = 0</code> , ZBTSRAM 1 is mapped to this region. Otherwise, <code>Remap_ctrl = 0</code> maps Block RAM and <code>Remap_ctrl = 1</code> maps ZBTSRAM 1. The V2M-MPS2 board microcontroller controls the <code>zbt_boot_ctrl</code> signal. The <code>zbt_boot_ctrl</code> signal overrides the boot option to enable use of the ZBT RAM.	Yes	0x00000000 to 0x00003FFF

7.3.2 MPS2 memory map for models with the Armv8-M additions

This section describes the MPS2 memory map for newer cores, with the Armv8-M additions.

For standard Arm peripherals, see the TRM for that device.



- A bus error is generated for accesses to memory areas not shown in this table.
- Any memory device that does not occupy the total region is aliased within that region.

Table 7-2: Memory map for models with the Armv8-M additions

Description	IDAU	Modeled	Address range
ZBTSRAM 1 (4MB) in Non-secure (NS) world. Reserved 8MB, only 4MB implemented. VTOR initialization value to be configurable in LAC. Second half (4MB) aliased to first half (4MB).	NS	Yes	0x00000000 to 0x007FFFFFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	N/A	0x00800000 to 0x0FFFFFFF
ZBTSRAM 1 (4MB) in Secure (S) world. Reserved 8MB, only 4MB implemented. Second half (4MB) aliased to first half (4MB).	S	Yes	0x10000000 to 0x107FFFFFFF
Not used. Default expansion port: bus error.	S	N/A	0x10800000 to 0x1FFFFFFF
ZBTSRAM 2 and ZBTSRAM 3 (4MB) in NS world. Reserved 8MB, only 4MB implemented. For IoT subsystems, different cores have different memory sizes. Second half (4MB) aliased to first half (4MB).	NS	Yes	0x20000000 to 0x207FFFFFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	N/A	0x20800000 to 0x20FFFFFFF
PSRAM (32MB)	NS	Yes	0x21000000 to 0x22FFFFFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	N/A	0x23000000 to 0x23FFFFFFF
MTB SRAM. Reserved 64KB, only 16KB implemented. Aliased to 0x0 for booting in RTL simulation.	NS	Yes	0x24000000 to 0x2400FFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	N/A	0x24010000 to 0x2FFFFFFF
ZBTSRAM 2 and ZBTSRAM 3 (4MB) in S world. Reserved 8MB, only 4MB implemented. Second half (4MB) aliased to first half (4MB).	S	Yes	0x30000000 to 0x307FFFFFFF
Not used. Default expansion port: bus error.	S	N/A	0x30800000 to 0x30FFFFFFF
Not used. No memory gating unit on PSRAM (16MB) path because it is shared with Ethernet control. Default expansion port: bus error.	S	N/A	0x31000000 to 0x31FFFFFFF
Not used. Default expansion port: bus error.	S	N/A	0x30800000 to 0x3FFFFFFF
Timer 0. Non-secure CMSDK APB subsystem.	NS	Yes	0x40000000 to 0x40000FFF
Timer 1. Non-secure CMSDK APB subsystem.	NS	Yes	0x40001000 to 0x40001FFF
Dual Timer. Non-secure CMSDK APB subsystem.	NS	Yes	0x40002000 to 0x40002FFF
Not used. Non-secure CMSDK APB subsystem.	NS	N/A	0x40003000 to 0x40003FFF
UART #0. Non-secure CMSDK APB subsystem.	NS	Yes	0x40004000 to 0x40004FFF
UART #1. Non-secure CMSDK APB subsystem.	NS	Yes	0x40005000 to 0x40005FFF
UART #2. Non-secure CMSDK APB subsystem.	NS	Yes	0x40006000 to 0x40006FFF

Description	IDAU	Modeled	Address range
Not used. Non-secure CMSDK APB subsystem.	NS	N/A	0x40007000 to 0x40007FFF
Watchdog. Non-secure CMSDK APB subsystem.	NS	Yes	0x40008000 to 0x40008FFF
Not used. Non-secure CMSDK APB subsystem.	NS	N/A	0x40009000 to 0x4000F000
GPIO #0.	NS	Yes	0x40010000 to 0x40010FFF
GPIO #1.	NS	Yes	0x40011000 to 0x40011FFF
GPIO #2.	NS	Yes	0x40012000 to 0x40012FFF
GPIO #3.	NS	Yes	0x40013000 to 0x40013FFF
Default slave inside AHB peripheral subsystem. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	Yes	0x40014000 to 0x40017FFF
DMA Controller #0.	NS	Yes	0x40018000 to 0x40018FFF
DMA Controller #1.	NS	Yes	0x40019000 to 0x40019FFF
Default slave inside AHB peripheral subsystem. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	Yes	0x4001A000 to 0x4001EFFF
CMSDK system controller. PMU control and remap registers unused. Only reset option (lockup reset) and rest info available.	NS	Yes	0x4001F000 to 0x4001FFFF
Not used. Non-secure FPGA APB subsystem (unused APB space: RAZ/WI).	NS	N/A	0x40020000 to 0x40020FFF
PL022 (SPI for LCD). Non-secure FPGA APB subsystem (unused APB space: RAZ/WI).	NS	Partial	0x40021000 to 0x40021FFF
SBCon I2C (Touch for LCD). Non-secure FPGA APB subsystem (unused APB space: RAZ/WI).	NS	Partial	0x40022000 to 0x40022FFF
SBCon I2C (Audio configuration). Non-secure FPGA APB subsystem (unused APB space: RAZ/WI).	NS	Yes	0x40023000 to 0x40023FFF
Audio I2S. Non-secure FPGA APB subsystem (unused APB space: RAZ/WI).	NS	Partial	0x40024000 to 0x40024FFF
Not used. Non-secure FPGA APB subsystem (unused APB space: RAZ/WI).	NS	N/A	0x40025000 to 0x40027FFF
FPGA system control & I/O (LEDs, buttons...). Non-secure FPGA APB subsystem (unused APB space: RAZ/WI).	NS	Yes	0x40028000 to 0x40028FFF
Not used. Non-secure FPGA APB subsystem (unused APB space: RAZ/WI).	NS	N/A	0x40029000 to 0x4002EFFF
SCC registers. Non-secure FPGA APB subsystem (unused APB space: RAZ/WI).	NS	Yes	0x4002F000 to 0x4002FFFF
Not used.	NS	N/A	0x40030000 to 0x40113FFF
SVOS DualTimer. Only enabled for Cortex-M55 SVOS.	NS	Yes	0x40114000 to 0x40114fff

Description	IDAU	Modeled	Address range
Not used.	NS	N/A	0x40115000 to 0x401FFFFFF
Ethernet (SMSC 91C111).	NS	Partial	0x40200000 to 0x402FFFFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	N/A	0x40300000 to 0x40FFFFFFF
VGA console.	NS	Yes	0x41000000 to 0x4100FFFF
Not used.	NS	N/A	0x41010000 to 0x410FFFFFF
VGA image.	NS	Yes	0x41100000 to 0x4113FFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	N/A	0x41140000 to 0x4FFFFFFF
Timer 0. Secure CMSDK APB subsystem.	S	Yes	0x50000000 to 0x50000FFF
Timer 1. Secure CMSDK APB subsystem.	S	Yes	0x50001000 to 0x50001FFF
Dual Timer. Secure CMSDK APB subsystem.	S	Yes	0x50002000 to 0x50002FFF
Not used. Secure CMSDK APB subsystem.	S	N/A	0x50003000 to 0x50003FFF
UART #0. Secure CMSDK APB subsystem.	S	Yes	0x50004000 to 0x50004FFF
UART #1. Secure CMSDK APB subsystem.	S	Yes	0x50005000 to 0x50005FFF
UART #2. Secure CMSDK APB subsystem.	S	Yes	0x50006000 to 0x50006FFF
Not used. Secure CMSDK APB subsystem.	S	N/A	0x50007000 to 0x50007FFF
Watchdog. Secure CMSDK APB subsystem.	S	Yes	0x50008000 to 0x50008FFF
Not used. Secure CMSDK APB subsystem.	S	N/A	0x50009000 to 0x5000F000
GPIO #0	S	Yes	0x50010000 to 0x50010FFF
GPIO #1	S	Yes	0x50011000 to 0x50011FFF
GPIO #2	S	Yes	0x50012000 to 0x50012FFF
GPIO #3	S	Yes	0x50013000 to 0x50013FFF
Default slave. Default expansion port: bus error.	S	Yes	0x50014000 to 0x50017FFF
DMA Controller #0.	S	Yes	0x50018000 to 0x50018FFF

Description	IDAU	Modeled	Address range
DMA Controller #1.	S	Yes	0x50019000 to 0x50019FFF
Default slave. Default expansion port: bus error.	S	Yes	0x5001A000 to 0x5001EFFF
CMSDK system controller. PMU control and remap registers unused. Only reset option (lockup reset) and rest info available.	S	Yes	0x5001F000 to 0x5001FFFF
FPGA APB subsystem	S	-	0x50020000 to 0x5002FFFF
Not used. Secure FPGA APB subsystem	S	N/A	0x50020000 to 0x50020FFF
PL022 (SPI for LCD). Secure FPGA APB subsystem	S	Partial	0x50021000 to 0x50021FFF
SBCon I2C (touch for LCD). Secure FPGA APB subsystem	S	Partial	0x50022000 to 0x50022FFF
SBCon I2C (audio configuration). Secure FPGA APB subsystem	S	Yes	0x50023000 to 0x50023FFF
Audio I2S. Secure FPGA APB subsystem	S	Partial	0x50024000 to 0x50024FFF
Not used. Secure FPGA APB subsystem	S	N/A	0x50025000 to 0x50027FFF
FPGA system control & I/O (LEDs, buttons...). Secure FPGA APB subsystem	S	Yes	0x50028000 to 0x50028FFF
Not used. Secure FPGA APB subsystem	S	N/A	0x50029000 to 0x5002EFFF
SCC registers. Secure FPGA APB subsystem	S	Yes	0x5002F000 to 0x5002FFFF
Not used.	S	N/A	0x50030000 to 0x50113FFF
SVOS DualTimer. Only enabled for Cortex-M55 SVOS.	S	Yes	0x50114000 to 0x50114fff
Not used.	S	N/A	0x50115000 to 0x501FFFFF
Ethernet (SMSC 91C111).	S	Partial	0x50200000 to 0x502FFFFF
Not used. Default expansion port: bus error.	S	N/A	0x50300000 to 0x50FFFFFF
VGA console.	S	Yes	0x51000000 to 0x5100FFFF
Not used.	S	N/A	0x51010000 to 0x510FFFFF
VGA image.	S	Yes	0x51100000 to 0x5113FFFF
Not used.	S	N/A	0x51140000 to 0x58006FFF
Secure Control Registers. Secure APB subsystem.	S	Yes	0x58007000 to 0x58007FFF

Description	IDAU	Modeled	Address range
Flash memory gating unit configuration (mapped to AHB port for CODE region in the bus matrix, not APB). Secure APB subsystem.	S	Yes	0x58008000 to 0x58009FFF
SRAM memory gating unit configuration (mapped to AHB port for SRAM region in the bus matrix, not APB). Secure APB subsystem.	S	Yes	0x5800A000 to 0x5800DFFF
Reserved. Secure APB subsystem.	S	N/A	0x5800E000 to 0x5800EFFF
Reserved. Secure APB subsystem.	S	N/A	0x5800F000 to 0x5800FFFF
Not used. Default expansion port: bus error.	S	N/A	0x58010000 to 0x5FFFFFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	N/A	0x60000000 to 0x6FFFFFFF
Not used. Default expansion port: bus error.	S	N/A	0x70000000 to 0x7FFFFFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	N/A	0x80000000 to 0x8FFFFFFF
Not used. Default expansion port: bus error.	S	N/A	0x90000000 to 0x9FFFFFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	N/A	0xA0000000 to 0xAFFFFFFF
Not used. Default expansion port: bus error.	S	N/A	0xB0000000 to 0xBFFFFFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	N/A	0xC0000000 to 0xCFFFFFFF
Not used. Default expansion port: bus error.	S	N/A	0xD0000000 to 0xDFFFFFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	N/A	0xE0000000 to 0xEFFFFFFF
System ROM table. Exempted from checking.	Exempt	Yes	0xF0000000 to 0xF0000FFF
Not used. Default expansion port: bus error.	Exempt	N/A	0xF0001000 to 0xF000FFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	N/A	0xF0100000 to 0xF01FFFFF
MTB SFR address space.	NS	Yes	0xF0200000 to 0xF0200FFF
Reserved. This region is non-executable.	NS	N/A	0xF0210000 to 0xF0213FFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI .	NS	N/A	0xF0214000 to 0xF0217FFF

7.4 MPS2 interrupt assignments

This section describes the interrupt assignments.

Table 7-3: Interrupt assignments

Number	Interrupt
NMI	Watchdog.
0	UART 0 receive interrupt.
1	UART 0 transmit interrupt.
2	UART 1 receive interrupt.
3	UART 1 transmit interrupt.
4	UART 2 receive interrupt.
5	UART 2 transmit interrupt.
6	GPIO 0, 2 combined interrupt.
7	GPIO 1, 3 combined interrupt.
8	Timer 0.
9	Timer 1.
10	Dual Timer.
11	SPI #1 (LCD). The LCD had shared SPI #0 and SPI #1.
12	UART overflow (0, 1, 2).
13	Ethernet.
14	Audio I2S.
15	Touch screen.
16-31	GPIO 0 individual interrupts.
32-47	GPIO 1 individual interrupts. ArmR8-M additions.
48	SPI #0. ArmR8-M addition.
49	Reserved.
50	TRNG (Secure). ArmR8-M addition.
51	Unique ID and Secure storage (Secure). ArmR8-M addition.
52	DMA controller #0.
53	DMA controller #1.
54	SecureErrorIRQ. ArmR8-M addition. Detection of Non-secure access to Secure address spaces (including other bus masters). Generated by Memory Gating unit, Peripheral Gating units, bus gasket for legacy bus masters.

7.5 MPS2 - differences between models and hardware

This section describes the features of the hardware that the models do not implement, or implement with significant differences.

MPS2 implements most devices. Some peripherals have minimal implementations:

- The Ethernet module in the model is a LAN91C111. The hardware provides a LAN9220.

- The Audio module is **RAZ/WI**.
- The STMPE811 touchscreen module only reports touch positions.
- The model of the Ampire LCD module supports a subset of the graphics modes.

You can display images and text on an emulated VGA output, images on the LCD, and text on the UART.

RX overrun mode

The CMSDK_UART component has a parameter `rx_overrun_mode` that controls how to handle the transfer of characters into the UART RX FIFO when the `RX_OVERRUN` flag is set in the STATE register. It has the following possible values:

0

This is the default value. Never block the transfer of the next character into the RX FIFO even if it means losing multiple characters. This is the same behavior as the hardware. It might be useful when evaluating software design, for example to indicate whether the UART can be serviced quickly enough.

In this mode, as in the hardware, the `RX_OVERRUN` flag serves only to alert software to the fact that characters have been lost and that some sort of corrective action, or recovery procedure might be required.

1

Pause the transfer of characters into the RX FIFO before any characters are lost. This makes the serial connection lossless even if the software does not make any attempt to service the UART in a timely fashion.

In this mode, the transfer of characters can be resumed by reading a character from the DATA register, so no special action is required by software.

2

Pause the transfer of characters into the UART RX FIFO after the first character is lost due to overrun.

In this mode, clearing the `RX_OVERRUN` flag resumes the transfer of characters. This requires the software to write to the STATE or INT register to clear the flag. Precisely one character will have been lost. This is the legacy behavior of the UART.

Arm®v8-M

The model does not support MTB, ETM, and TPIU. MTB RAM is absent.

In the Memory Gating Unit, the model provides a configurable block size. For performance reasons, the minimum block size in the model is 4096 bytes. Hardware and later models might allow smaller block sizes. Software must use the `BLK_CFG` register to determine block size.

Timing

FVPs enable software applications to run in a functionally accurate simulation. However, because of the relative balance of fast simulation speed over timing accuracy, there are situations where the models might behave unexpectedly.

If your code interacts with real world devices such as timers and keyboards, data arrives in the modeled device in real world, or wall clock, time. However, simulation time can run faster than the wall clock. So, a single key press might be interpreted as several repeated key presses, or a single mouse click might be interpreted as a double click.

To avoid this mismatch, the FVPs provide the Rate Limit feature. Enabling Rate Limit forces the model to run at wall clock time. For interactive applications, Arm recommends enabling Rate Limit. Use the Rate Limit button in the CLCD display or the `rate_limit-enable` model instantiation parameter.

8. Versatile Express Model

This chapter describes the components of Fast Models that are specific to the Versatile™ Express (VE) model of the hardware platform.

8.1 VE

The Versatile Express (VE) FVPs are functionally-accurate system models for software execution. A range of VE FVPs are supplied as standalone products and as examples in Fast Models.

For a list of the VE Platform FVPs that are included in the FVP Standard Library package, see [VE Platform FVPs](#) in the FVP Reference Guide. The package can be downloaded from [Fixed Virtual Platforms](#).

LISA+ source and project files for the VE FVPs are available in the `$PVLIB_HOME/examples/LISA/FVP_VE/` directory.

Arm produces the VE hardware development platform. The Motherboard Express µAdvanced Technology Extended (ATX) V2M-P1 is the basis for an integrated software and hardware development system. This system is also based on the Arm Symmetric MultiProcessor (SMP) system architecture.

The VE FVPs are system models implemented in software. Each model contains:

- A virtual implementation of the Arm VE motherboard.
- A single daughterboard containing one or more Arm processors.
- Associated interconnections.

The motherboard provides:

- Peripherals for multimedia or networking environments.
- Access to motherboard peripherals and functions through a static memory bus to simplify access from daughterboards.
- High-performance PCI-Express slots for expansion cards.
- Consistent memory maps with different processor daughterboards that simplify software development and porting.
- Automatic detection and configuration of attached CoreTile Express and LogicTile Express daughterboards.
- Automatic shutdown for over-temperature or power supply failure.
- No system power-on for unconfigurable daughterboards.
- Power sequencing of system.
- Drag and drop file updating of configuration files.
- Support of either a 12V power-supply unit or an external ATX power supply.

- Support of FPGA and processor daughterboards to provide custom peripherals, early access to processor designs, or production test chips.

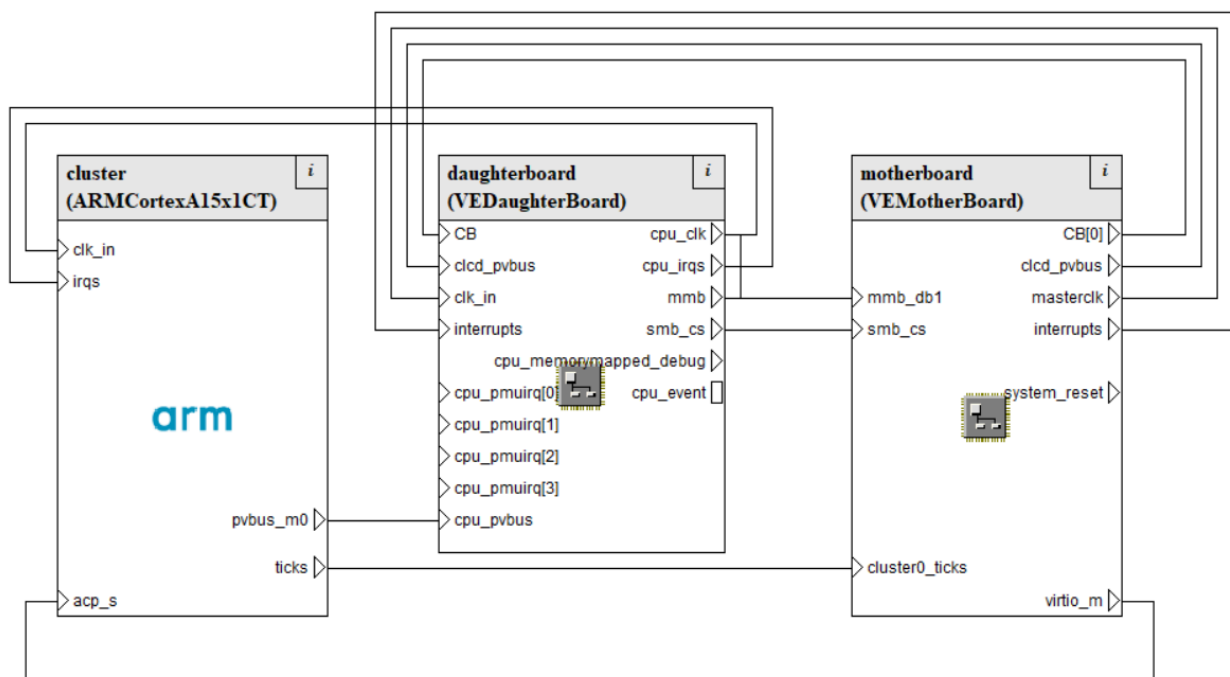


Arm bases the models on the VE platform memory map, but does not intend them to be accurate representations of a specific VE hardware revision. The VE FVPs support selected peripherals. The models are sufficiently complete and accurate to boot the same operating system images as the VE hardware.

VE FVPs provide functionally accurate models for software execution. However, the models sacrifice timing accuracy to increase simulation speed. Key deviations from hardware are:

- Approximate timing.
- Simplified buses.
- No implementations for processor caches and the related write buffers.

Figure 8-1: Top-level block diagram of a VE model



Related information

- [Getting Started with Fixed Virtual Platforms](#)
- [ARM Motherboard Express µATX V2M-P1 Technical Reference Manual](#)

8.2 VE memory map for Cortex-A series

The global memory map for the Cortex®-A series VE model is based on the hardware Versatile™ Express RS1 memory map with the RS2 extensions.



Note

The VE FVP implementation of memory does not require the memory controller to have the correct values. If you run applications on hardware, ensure that the memory controller is set up properly. Otherwise, applications that run on the FVP might fail on hardware.

Table 8-1: Cortex®-A series platform model memory map

Name	Modeled	Address range	Size
NOR FLASH0 (CS0)	Yes	0x00_00000000-0x00_03FFFFFF	64MB
Reserved	-	0x00_04000000-0x00_07FFFFFF	64MB
NOR FLASH0 alias (CS0)	Yes	0x00_08000000-0x00_0BFFFFFF	64MB
NOR FLASH1 (CS4)	Yes	0x00_0C000000-0x00_0FFFFFFF	64MB
Unused (CS5)	-	0x00_10000000-0x00_13FFFFFF	-
PSRAM (CS1) - unused	No	0x00_14000000-0x00_17FFFFFF	-
Peripherals (CS2). See Table: CS2 region peripheral memory map, below.	Yes	0x00_18000000-0x00_1BFFFFFF	64MB
Peripherals (CS3). See Table: CS2 region peripheral memory map, below.	Yes	0x00_1C000000-0x00_1FFFFFFF	64MB
CoreSight™ and peripherals	No	0x00_20000000-0x00_2CFFFFFF. See Note after table.	-
Graphics space	No	0x00_2D000000-0x00_2D00FFFF	-
System SRAM	Yes	0x00_2E000000-0x00_2EFFFFFF	64KB
Ext AXI	No	0x00_2F000000-0x00_7FFFFFFF	-
4GB DRAM (in 32-bit address space). See Note after table.	Yes	0x00_80000000-0x00_FFFFFFFF	2GB
Unused	-	0x01_00000000-0x07_FFFFFFFF	-
4GB DRAM (in 36-bit address space). See Note after table.	Yes	0x08_00000000-0x08_FFFFFFFF	4GB
Unused	-	0x09_00000000-0x7F_FFFFFFFF	-
4GB DRAM (in 40-bit address space). See Note after table.	Yes	0x80_00000000-0xFF_FFFFFFFF	4GB



Note

- The private peripheral address 0x2c000000 is mapped in the CoreSight™ and peripherals region. Use the parameter PERIPHBASE to map the peripherals to a different address.
- The model contains a single 4GB block of DRAM, which is aliased across the three different DRAM regions. In other words, it can be accessed at three different physical addresses, which are all mapped to the same area of DRAM. For example, a write to address 0x00_80000000 is visible to reads at address 0x80_00000000. The lowest of the physical address regions is only 2GB in size.

The model has a `secure_memory` option. When you enable this option, the memory map changes for a number of peripherals.

Table 8-2: CS2 region peripheral memory map for `secure_memory` option

Peripheral	Address range	Functionality with <code>secure_memory</code> enabled
NOR FLASH0 (CS0)	0x00_00000000-0x00_0001FFFF	Secure RO, aborts on non-secure accesses.
Reserved	0x00_04000000-0x00_0401FFFF	Secure SRAM, aborts on non-secure accesses.
NOR FLASH0 alias (CS0)	0x00_08000000-0x00_7DFFFFFF	Normal memory map, aborts on secure accesses.
Ext AXI	0x00_7e000000-0x00_7FFFFFFF	Secure DRAM, aborts on non-secure accesses.
4GB DRAM (in 32-bit address space)	0x00_80000000-0xFF_FFFFFFFF	Normal memory map, aborts on secure accesses.

Table 8-3: CS2 region peripheral memory map

Peripheral	Modeled	Address range	Size	GIC Int. See Note after table.
VRAM - aliased	Yes	0x00_18000000-0x00_19FFFFFF	32MB	-
Ethernet (SMSC 91C111)	Yes	0x00_1A000000-0x00_1AFFFFFF	16MB	47
USB - unused	No	0x00_1B000000-0x00_1BFFFFFF	16MB	-



Use these interrupt signal values to program your interrupt controller. They are the SPI number plus 32. Add 32 to the interrupt number from the peripherals to form the interrupt number that the GIC sees. GIC interrupts 0-31 are for internal use.

Table 8-4: CS3 region peripheral memory map

Peripheral	Modeled	Address range	Size	GIC Int. See Note after table.
Local DAP ROM	No	0x00_1C000000-0x00_1C00FFFF	64KB	-
VE System Registers	Yes	0x00_1C010000-0x00_1C01FFFF	64KB	-
System Controller (SP810)	Yes	0x00_1C020000-0x00_1C02FFFF	64KB	-
TwoWire serial interface (PCIe)	No	0x00_1C030000-0x00_1C03FFFF	64KB	-
AACI (PL041)	Yes	0x00_1C040000-0x00_1C04FFFF	64KB	43
MCI (PL180)	Yes	0x00_1C050000-0x00_1C05FFFF	64KB	41, 42
KMI - keyboard (PL050)	Yes	0x00_1C060000-0x00_1C06FFFF	64KB	44
KMI - mouse (PL050)	Yes	0x00_1C070000-0x00_1C07FFFF	64KB	45
Reserved	-	0x00_1C080000-0x00_1C08FFFF	64KB	-
UART0 (PL011)	Yes	0x00_1C090000-0x00_1C09FFFF	64KB	37
UART1 (PL011)	Yes	0x00_1C0A0000-0x00_1C0AFFFF	64KB	38
UART2 (PL011)	Yes	0x00_1C0B0000-0x00_1C0BFFFF	64KB	39
UART3 (PL011)	Yes	0x00_1C0C0000-0x00_1C0CFFFF	64KB	40
Reserved	-	0x00_1C0D0000-0x00_1C0EFFFF	128KB	-
Watchdog (SP805)	Yes	0x00_1C0F0000-0x00_1C0FFFFFFF	64KB	32
Reserved	-	0x00_1C100000-0x00_1C10FFFF	64KB	-
Timer-0 (SP804)	Yes	0x00_1C110000-0x00_1C11FFFF	64KB	34
Timer-1 (SP804)	Yes	0x00_1C120000-0x00_1C12FFFF	64KB	35

Peripheral	Modeled	Address range	Size	GIC Int. See Note after table.
Virtio block device	Yes	0x00_1C130000-0x00_1C13FFFF	64KB	74
Virtio P9 device	Yes	0x00_1C140000-0x00_1C14FFFF	64KB	75
Reserved	-	0x00_1C130000-0x00_1C15FFFF	192KB	-
TwoWire serial interface (DVI) - unused	No	0x00_1C160000-0x00_1C16FFFF	64KB	-
Real-time Clock (PL031)	Yes	0x00_1C170000-0x00_1C17FFFF	64KB	36
Reserved	-	0x00_1C180000-0x00_1C19FFFF	128KB	-
CF Card - unused	No	0x00_1C1A0000-0x00_1C1AFFFF	64KB	-
Reserved	-	0x00_1C1B0000-0x00_1C1EFFFF	256KB	-
Color LCD Controller (PL111)	Yes	0x00_1C1F0000-0x00_1C1FFFFF	64KB	46
Reserved	-	0x00_1C200000-0x00_1FFFFFFF	64KB	-



Use these interrupt signal values to program your interrupt controller. They are the SPI number plus 32. Add 32 to the interrupt number from the peripherals to form the interrupt number that the GIC sees. GIC interrupts 0-31 are for internal use.

8.3 VE memory map for Cortex-R series

The Versatile™ Express RS1 memory map with the RS2 extensions is the base of the global memory map for the Cortex®-R series platform model.

Table 8-5: Cortex®-R series VE FVP memory map

Memory	Modeled	Address range
DRAM	Yes	0x00000000-0x3FFFFFFF
FLASH0	Yes	0x40000000-0x43FFFFFF
FLASH1	Yes	0x44000000-0x47FFFFFF
PSRAM	Yes	0x48000000-0x4BFFFFFF
RAM	No	0x4C000000-0x4FFFFFFF
PL390 GIC CPU Interface. Cortex®-R4 and Cortex®-R5 models only.	Yes	0xAE000000-0xAE000FFF
PL390 GIC Distributor. Cortex®-R4 and Cortex®-R5 models only.	Yes	0xAE001000-0xAE001FFF
VE System Registers	Yes	0xB0000000-0xB000FFFF
SP810	Yes	0xB0010000-0xB001FFFF
PL041 AACI	Yes	0xB0040000-0xB004FFFF
PL180 MCI	Yes	0xB0050000-0xB005FFFF
PL050 KMIO	Yes	0xB0060000-0xB006FFFF
PL050 KMI1	Yes	0xB0070000-0xB007FFFF
PL011 UART0	Yes	0xB0090000-0xB009FFFF
PL011 UART1	Yes	0xB00A0000-0xB00AFFFF
PL011 UART2	Yes	0xB00B0000-0xB00BFFFF
PL011 UART3	Yes	0xB00C0000-0xB00CFFFF

Memory	Modeled	Address range
SP805 WATCHDOG	Yes	0xB00F0000-0xB00FFFFF
TIMER_0_1	Yes	0xB0110000-0xB011FFFF
TIMER_2_3	Yes	0xB0120000-0xB012FFFF
PL031 Real Time Clock	Yes	0xB0170000-0xB017FFFF
Compact Flash	No	0xB01A0000-0xB01AFFFF
PL011 UART4	Yes	0xB01B0000-0xB01BFFFF
PL111 CLCD	Yes	0xB01F0000-0xB01FFFFFF. See Note after table
RAM	No	0xB4000000-0xBBFFFFFF
Video RAM	Yes	0xBC000000-0xBDFFFFFF
Ethernet (SMSC 91C111)	Yes	0xBE000000-0xBEFFFFFF
USB	No	0xBF000000-0xBFFFFFFF



For Cortex®-R4 and Cortex®-R5 models, the range for PL111 CLCD is 0xA0000000-0xA0010000

8.4 VE interrupt assignments for Cortex-A series

The platform routes the following Shared Peripheral Interrupts (SPIs) to the GIC.

Table 8-6: SPI GIC assignments

IRQ ID	SPI offset	Device
32	0	Watchdog, SP805
34	2	Dual timer 0/1, SP804
35	3	Dual timer 2/3, SP804
36	4	Real-time Clock, PL031
37	5	UART0, PL011
38	6	UART1, PL011
39	7	UART2, PL011
40	8	UART3, PL011
41	9	MCI, PL180, MCIINTRO
42	10	MCI, PL180, MCIINTR1
43	11	AACI, PL041
44	12	KMI - Keyboard, PL050
45	13	KMI - Mouse, PL050
46	14	Color LCD Controller, PL111
47	15	Ethernet, SMSC 91C111
74	42	Virtio block device
75	43	Virtio P9 device

IRQ ID	SPI offset	Device
92	60	CPU 0 PMU
93	61	CPU 1 PMU
94	62	CPU 2 PMU
95	63	CPU 3 PMU
117	85	HDLCD

8.5 VE interrupt assignments for Cortex-R series

These tables describe the interrupt assignments for Cortex-R4, Cortex-R5, Cortex-R7 and Cortex-R8.

Table 8-7: Interrupt assignments for Cortex-R4 and Cortex-R5

GIC IRQ ID	SPI offset	Device
32	0	Watchdog, SP805
34	2	Dual timer 0/1, SP804
35	3	Dual timer 2/3, SP804
36	4	Real-time Clock, PL031
41	9	MCI, PL180, MCIINTRO
42	10	MCI, PL180, MCIINTR1
43	11	AACI, PL041
44	12	KMI - Keyboard, PL050
45	13	KMI - Mouse, PL050
47	15	Ethernet, SMSC 91C111
75	16	Level 2 Cache Controller, PL310 Combined interrupt
76	14	Color LCD Controller, PL111
96	5	UART0, PL011
97	6	UART1, PL011
98	7	UART2, PL011
99	8	UART3, PL011

Table 8-8: Interrupt assignments for Cortex-R7 and Cortex-R8

GIC IRQ ID	SPI offset	Device
32	0	Watchdog, SP805
34	2	Dual timer 0/1, SP804
35	3	Dual timer 2/3, SP804
36	4	Real-time Clock, PL031
37	5	UART0, PL011
38	6	UART1, PL011
39	7	UART2, PL011
40	8	UART3, PL011

GIC IRQ ID	SPI offset	Device
41	9	MCI, PL180, MCIINTR0
42	10	MCI, PL180, MCIINTR1
43	11	AACI, PL041
44	12	KMI - Keyboard, PL050
45	13	KMI - Mouse, PL050
46	14	Color LCD Controller, PL111
47	15	Ethernet, SMSC 91C111

8.6 VE parameters

This section describes the VE FVP instantiation parameters.

8.6.1 VE instantiation parameters

This table describes the instantiation parameters for VE models.

Table 8-9: VE instantiation parameters

Component	Parameter	Type	Allowed values	Default value	Description
ve_sysregs	user_switches_value	Integer	See VE switch S6 .	0	Switch S6 setting.
flashloader0	fname	String	Valid filename	""	Path to flash image file.
flashloader1	fname	String	Valid filename	""	Path to flash image file.
mmc	p_mmc_file	String	Valid filename	mmc.dat	Multimedia card filename.
pl111_clcd	pixel_double_limit	Integer	-	12c	Sets threshold in horizontal pixels below which pixels sent to framebuffer are doubled in size in both dimensions.
sp810_sysctrl	use_s8	Boolean	true, false	false	Indicates whether to read boot_switches_value.

8.6.2 VE secure memory parameters

This table describes the VE FVP secure memory parameters that you can change when you start the model.

Table 8-10: VE secure memory parameters

Name	Type	Allowed values	Default value	Description
daughterboard.secure_memory	Boolean	true, false	false	<p>false</p> <p>The platform behaves as before.</p> <p>true</p> <p>The address space is segregated according to the security mode of the core. Some memory blocks near the bottom of the address space are available to Secure transactions only. The rest of the address space is available to Non-secure transactions only.</p>

8.6.3 VE switch S6

This section describes the behavior and default positions of the VE system model switch.

Switch S6 is equivalent to the Boot Monitor configuration switch on the VE hardware.

If you have the standard Arm® Boot Monitor flash image loaded, the setting of switch S6-1 changes what happens on model reset. Otherwise, the function of switch S6 is implementation dependent.

To write the switch position directly to the S6 parameter in the model, you must convert the switch settings to an integer value from the equivalent binary, where 1 is on and 0 is off.

Table 8-11: Default positions of VE system model switch

Switch	Default position	Function in default position
S6-1	OFF	Displays prompt permitting Boot Monitor command entry after system start.
S6-2	OFF	See STDIO redirection, below.
S6-3	OFF	See STDIO redirection, below.
S6-4 to S6-8	OFF	Reserved for application use.

If S6-1 is in the ON position, the Boot Monitor executes the boot script that was loaded into flash. If there is no script, the Boot Monitor prompt is displayed.

The settings of S6-2 and S6-3 affect STDIO source and destination on model reset.

Table 8-12: STDIO redirection

S6-2	S6-3	Output	Input	Description
OFF	OFF	UART0	UART0	STDIO autodetects whether to use semihosting I/O or a UART. If a debugger is connected, STDIO is redirected to the debugger output window, otherwise STDIO goes to UART0.
OFF	ON	UART0	UART0	STDIO is redirected to UART0, regardless of semihosting settings.

S6-2	S6-3	Output	Input	Description
ON	OFF	CLCD	Keyboard	STDIO is redirected to the CLCD and keyboard, regardless of semihosting settings.
ON	ON	CLCD	UART0	STDIO output is redirected to the LCD and input is redirected to UART0, regardless of semihosting settings.

8.7 VE components

A complete model implementation of the VE platform includes both VE-specific components and generic components, such as buses and timers.

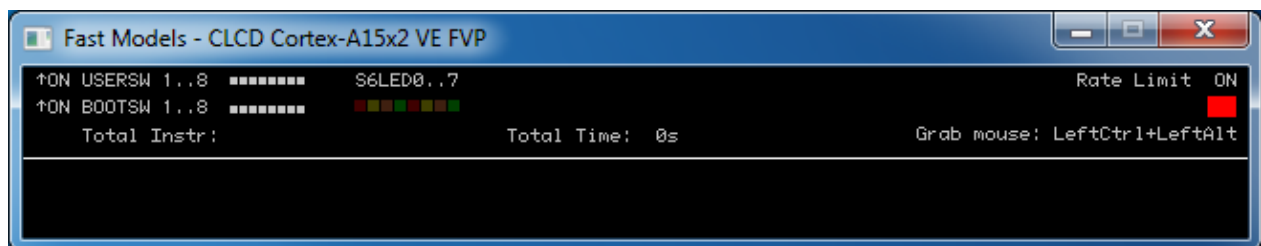
To see a list of all the component instances in the model, run it with the `--list-instances` option.

The generic components are documented in Fast Models components.

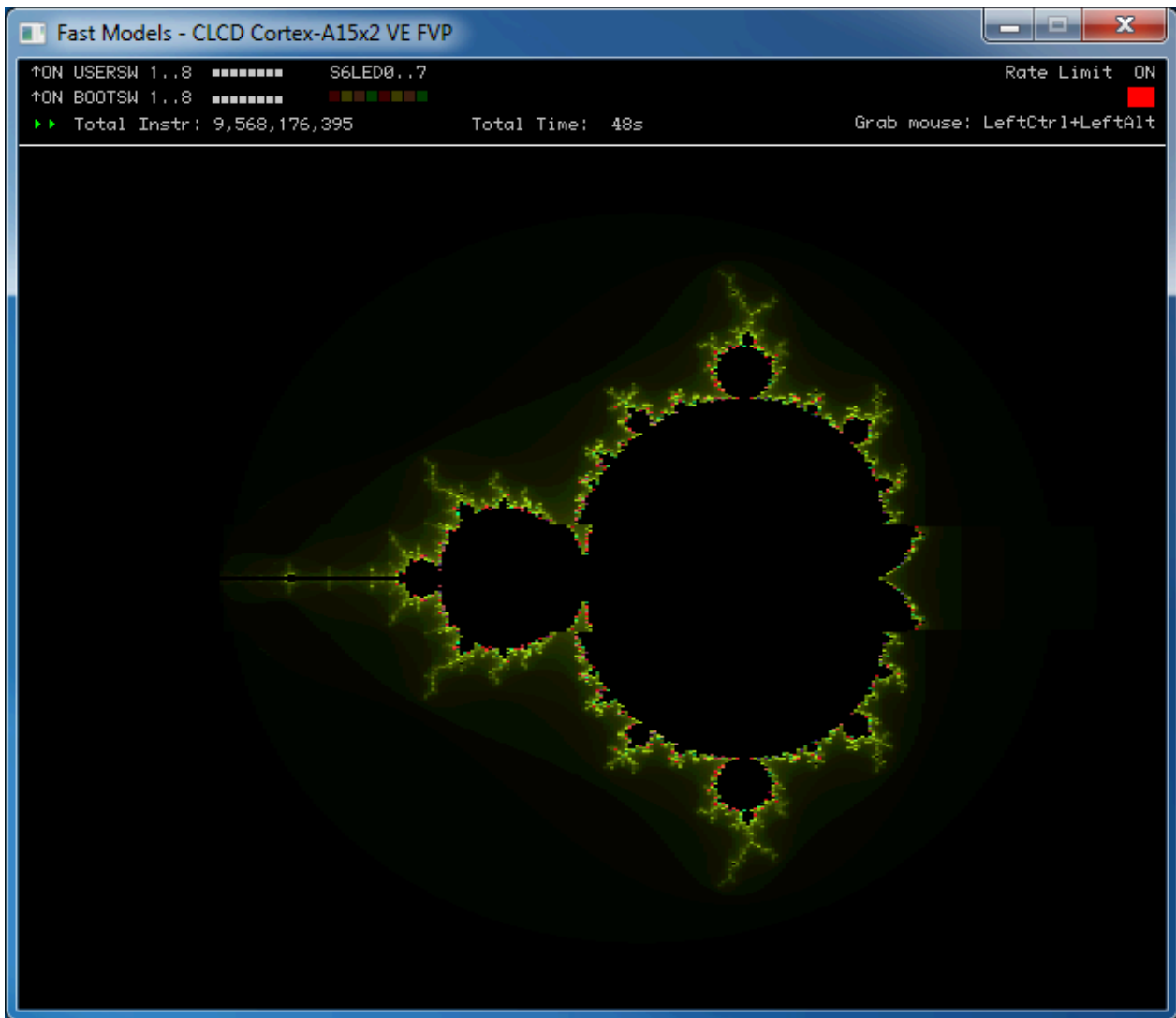
8.7.1 VEVisualisation component

This component can generate events from the host mouse or keyboard when the visualization window is in focus. For example, you can toggle the switch elements from the visualization window.

Figure 8-2: VE FVP CLCD visualization window



When a suitable application or system image loads, and configures the PL111_CLCD controller registers, the window expands to show the contents of the framebuffer.

Figure 8-3: VE FVP CLCD with brot.axf image

The VESimulation LISA+ component can be found in the `$PVLIB_HOME/examples/LISA/FVP_VE/LISA/` directory.



Note

Using this component can reduce simulation performance. Use the `rate_limit-enable` parameter to control simulation speed.

8.7.1.1 VEEVisualisation ports

This section describes the VEEVisualisation component ports.

Table 8-13: VEEVisualisation ports

Name	Protocol	Type	Description
boot_switch	ValueState	Slave	Provides state for the eight Boot DIP switches on the right side of the CLCD status bar.
clock_50Hz	ClockSignal	Slave	50Hz clock input.
cluster0_ticks[4]	InstructionCount	Slave	Connection from processor model in cluster 0 to show its current instruction count.
cluster1_ticks[4]	InstructionCount	Slave	Connection from processor model in cluster 1 to show its current instruction count.
daughter_leds	ValueState	Slave	A read/write port to read and set the value of the LEDs. 1 bit per LED, LSB left-most, up to 32 LEDs available. The LEDs appear only when parameter <code>daughter_led_count</code> is set to nonzero.
daughter_user_switches	ValueState	Slave	A read port to return the value of the daughter user switches. Write to this port to set the value of the switches, and use during reset only. LSB is left-most, up to 32 switches available.
keyboard	KeyboardStatus	Master	Output port providing key change events when the visualization window is in focus.
lcd	LCD	Slave	Connection from a CLCD controller for visualization of the frame buffer.
lcd_layout	LCDLayoutInfo	Master	Layout information for alphanumeric LCD display.
leds	ValueState	Slave	Displays state using the eight colored LEDs on the status bar.
mouse	MouseStatus	Master	Output port providing mouse movement and button events when the visualization window is in focus.
touch_screen	MouseStatus	Master	Provides mouse events when the visualization window is in focus.
user_switches	ValueState	Slave	Provides state for the eight User DIP switches on the left side of the CLCD status bar, equivalent to switch S6 on VE hardware.

8.7.1.2 VEEVisualisation parameters

This section describes the configuration parameters.

The syntax to use in a configuration file or on the command line is:

```
motherboard.vis.<parameter>=<value>
```



Note

Setting the `rate_limit-enable` parameter to `true` prevents the simulation from running too fast on fast workstations and enables timing loops and mouse actions to work correctly. However, it reduces the overall simulation speed. If your priority is high simulation speed, set `rate_limit-enable` to `false`, which is the default.

Table 8-14: VEEVisualisation parameters

Name	Type	Allowed values	Default value	Description
cluster0_name	string	-	Cluster0	Label for cluster 0 performance values.

Name	Type	Allowed values	Default value	Description
cluster1_name	string	-	Cluster1	Label for cluster 1 performance values.
cluster2_name	string	-	Cluster2	Label for cluster 2 performance values.
cluster3_name	string	-	Cluster3	Label for cluster 3 performance values.
cpu_name	string	-	-	Processor name displayed in window title.
daughter_led_count	int	0-32	0	Set to nonzero to display up to 32 LEDs. See the <code>daughter_leds</code> port.
daughter_user_switch_count	int	0-32	0	Set this parameter to display up to 32 switches. See the <code>daughter_user_switches</code> port.
diagnostics	bool	true, false	false	Print diagnostic messages.
disable_visualisation	bool	true, false	false	Disable the VEVisualisation component on model startup.
rate_limit-enable	bool	true, false	false	Restrict simulation speed so that simulation time more closely matches real time rather than running as fast as possible.
recorder.checkInstructionCount	bool	true, false	true	Check instruction count in recording file against actual instruction count during playback.
recorder.playbackFileName	string	-	""	Playback filename (empty string disables playback).
recorder.recordingFileName	string	-	""	Recording filename (empty string disables recording).
recorder.recordingTimeBase	int	-	0x5F5E100	Timebase in 1/s (relative to the master clock where 100000000 means 10 nanoseconds resolution simulated time for a 1Hz master clock) for recording. Higher values give higher time resolution, playback timebase is always taken from the playback file.
recorder.verbose	int	-	0	Enable verbose messages (1=normal, 2=even more).
trap_key	int	Valid ATKeyCode key value. See <code>\$PVLIB_HOME/include/components/KeyCode.h</code> for a list of values.	0x4A	Trap key that works with left Ctrl key to toggle mouse display. The default is the left Alt key, so press Left Alt and Left Ctrl simultaneously to toggle the mouse display.
window_title	string	-	"Fast Models - CLCD %cpu%"	Window title. <code>cpu_name</code> replaces <code>%cpu%</code> .

8.7.1.3 VEVisualisation verification and testing

This component passes tests by use as an I/O device for booting Linux and other operating systems.

8.7.1.4 VEEVisualisation performance

Arm expects the elements in the status bar to have little effect on the performance of PV systems. However, applications that often redraw the contents of the frame buffer might incur overhead through GUI interactions on the host OS.

8.7.1.5 VEEVisualisation library dependencies

This component relies on the Simple DirectMedia Layer (SDL) libraries, specifically `libSDL2-2.0.so.0.10.0`.

This library is bundled with Fast Models and is also available as an rpm for Red Hat Enterprise Linux. On Windows, the library is called `SDL2.dll`.

Related information

[Simple DirectMedia Layer \(SDL\) cross-platform development library](#)

8.8 VE_SysRegs component

This section describes the VE system registers component.

8.8.1 VE_SysRegs

This LISA+ component is a model of the VE status and system control registers.

8.8.2 VE_SysRegs ports

This table describes the VE_SysRegs ports.

Table 8-15: VE_SysRegs ports

Name	Protocol	Type	Description
<code>cb[0-1]</code>	<code>VECBProtocol</code>	Master	The Configuration Bus (CB) controls the power and reset sequence.
<code>clock_24Mhz</code>	<code>ClockSignal</code>	Slave	Reference clock for internal counter register.
<code>clock_100Hz</code>	<code>ClockSignal</code>	Slave	Reference clock for internal counter register.
<code>clock_CLCD</code>	<code>ClockRateControl</code>	Master	The clock for the LCD controller.
<code>lcd</code>	<code>LCD</code>	Master	Multimedia bus interface output to the LCD.
<code>leds</code>	<code>ValueState</code>	Master	Displays state of the <code>SYS_LED</code> register using the eight colored LEDs on the status bar.
<code>mmb[0-2]</code>	<code>LCD</code>	Slave	Multimedia bus interface input.
<code>mmc_card_present</code>	<code>StateSignal</code>	Slave	Indicates the presence of a MultiMedia Card (MMC) image.
<code>pvbuss</code>	<code>PVBus</code>	Slave	Slave port for connection to PV bus master/decoder.

Name	Protocol	Type	Description
system_reset	Signal	Master	Signal to the platform a complete system reset. Writes to the System Configuration registers can trigger the reset signal.
user_switches	ValueState	Master	Provides state for the eight User DIP switches on the left side of the CLCD status bar, equivalent to switch S6 on VE hardware.

8.8.3 VE_SysRegs parameters

This table describes the VE_SysRegs parameters.

Table 8-16: VE_SysRegs parameters

Name	Type	Default value	Description
exit_on_shutdown	bool	false	Used to shut down the system. See the Note after the table.
mmbSiteDefault	int	1	Default MultiMedia Bus (MMB) source (0=motherboard, 1=daughterboard 1, 2=daughterboard 2).
sys_proc_id0	int	0x0c000000	Processor ID register at CoreTile Express Site 1.
sys_proc_id1	int	0xff000000	Processor ID at CoreTile Express Site 2.
tilePresent	bool	true	Tile fitted.
user_switches_value	int	0	User switches.



When `exit_on_shutdown` is `true`, if software uses the `sys_cfgctrl` function `sys_cfg_shutdown`, then the simulator shuts down and exits. For more information on the `sys_cfgctrl` function values, see the Motherboard Express µATX V2M-P1 Technical Reference Manual.

8.8.4 VE_SysRegs registers

This table describes the configuration registers.

Table 8-17: VE_SysRegs registers

Name	Offset	Access	Description
SYS_ID	0x00	Read/write	System identity
SYS_SW	0x04	Read/write	Bits[7:0] map to switch S6
SYS_LED	0x08	Read/write	Bits[7:0] map to user LEDs
SYS_100HZ	0x24	Read only	100Hz counter
SYS_FLAGS	0x30	Read/write	General purpose flags
SYS_FLAGSCLR	0x34	Write only	Clear bits in general purpose flags
SYS_NVFLAGS	0x38	Read/write	General purpose non-volatile flags
SYS_NVFLAGSCLR	0x3C	Write only	Clear bits in general purpose non-volatile flags
SYS_MCI	0x48	Read only	MCI
SYS_FLASH	0x4C	Read/write	Flash control

Name	Offset	Access	Description
SYS_CFGSW	0x58	Read/write	Boot select switch
SYS_24MHZ	0x5C	Read only	24MHz counter
SYS_MISC	0x60	Read/write	Miscellaneous control flags
SYS_DMA	0x64	Read/write	DMA peripheral map
SYS_PROCID0	0x84	Read/write	Processor ID
SYS_PROCID1	0x88	Read/write	Processor ID
SYS_CFGDATA	0xA0	Read/write	Data to be read/written from/to motherboard controller
SYS_CFGCTRL	0xA4	Read/write	Control data transfer to motherboard controller
SYS_CFGSTAT	0xA8	Read/write	Status of data transfer to motherboard

8.8.5 VE_SysRegs - verification and testing

This component was tested as part of the Versatile™ Express model.

8.9 Differences between the VE hardware and the system models

This section describes features of the hardware that the models do not implement, or implement with significant differences.

Related information

[Fast Models accuracy](#) on page 23

[Processor implementation](#) on page 28

8.9.1 Memory map

The model represents the memory map of the hardware VE platform, but is not an accurate representation of a specific revision.

The memory map in the supplied model is sufficiently complete and accurate to boot the same operating system images as for the VE hardware.

In the memory map, memory regions that are not explicitly occupied by a peripheral or by memory are unmapped. This includes regions otherwise occupied by a peripheral that is not implemented, and those areas that are documented as reserved. Accessing these regions from the host processor results in the model presenting a warning.

8.9.2 Memory aliasing

The model implements address-space aliasing of the DRAM. This means that the same physical memory locations are visible at different addresses.

The lower 2GB of the DRAM is accessible at 0x00_80000000. The full 4GB of DRAM is accessible at 0x08_00000000 and again at 0x80_00000000. The aliasing of DRAM then repeats from 0x81_00000000 up to 0xFF_FFFFFFFF.

8.9.3 VE hardware features absent

These FVPs do not implement the following features of the hardware:

- Two-wire serial bus interfaces.
- USB interfaces.
- PCI Express interfaces.
- Compact flash.
- Digital Visual Interface (DVI).
- Debug and test interfaces.
- Dynamic Memory Controller (DMC).
- Static Memory Controller (SMC).

Related information

[VE memory map for Cortex-A_series](#) on page 4837

8.9.4 VE hardware features different

These Fixed Virtual Platforms only partially implement some features of the hardware.

The partially implemented features might not work as you expect. Check the model release notes for the latest information.

Sound

The VE FVPs implement the PL041 AACI PrimeCell and the audio CODEC as in the VE hardware, but with a limited number of sample rates.

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
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.
1123-00	13 September 2023	Non-Confidential	Update for v11.23.
1122-00	14 June 2023	Non-Confidential	Update for v11.22.

Issue	Date	Confidentiality	Change
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.
1107-00	17 May 2019	Non-Confidential	Update for v11.7.
1106-01	25 March 2019	Non-Confidential	Update for v11.6.1.

Issue	Date	Confidentiality	Change
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 technical changes to this documentation, 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.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
<i>italic</i>	Citations.
bold	Interface elements, such as menu names. Terms in descriptive lists, where appropriate.

Convention	Use
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, <Rd>, <CRn>, <CRm>, <Opcode_2></div>
SMALL CAPITALS	Terms that have specific technical meanings as defined in the <i>Arm® Glossary</i> . For example, IMPLEMENTATION DEFINED , IMPLEMENTATION SPECIFIC , UNKNOWN , and UNPREDICTABLE .



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.



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.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at developer.arm.com/documentation. Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
A-Profile Architecture	–	Non-Confidential
AMBA-PV Extensions to TLM User Guide	100962	Non-Confidential
AN521 Example SSE-200 Subsystem for MPS2+ Application Note	DAI 0521	Non-Confidential
Arm® Architecture Models	–	Non-Confidential
Arm® Corstone™ SSE-300 Example Subsystem Technical Reference Manual	101773	Non-Confidential
Arm® Cortex®-M23 Armv8-M IoT Kit User Guide	ECM 0635473	Non-Confidential
Arm® Cortex®-M33 processor Armv8-M IoT Kit FVP User Guide	ECM 0601256	Non-Confidential
Arm® Cortex®-M7 SMM on V2M-MPS2 Application Note 400	DAI 0400	Non-Confidential
Arm® DynamIQ Shared Unit Technical Reference Manual	100453	Non-Confidential
Arm® Socrates™	–	Non-Confidential
ASTFplugin FAQs	108074	Non-Confidential
Component Architecture Debug Interface User Guide	100963	Non-Confidential
Fast Models Fixed Virtual Platforms (FVP) Reference Guide	100966	Non-Confidential
Fast Models Model Trace Interface Reference Manual	DUI 0819	Non-Confidential
Fast Models User Guide	100965	Non-Confidential
Fixed Virtual Platforms	–	Non-Confidential
How to generate ASTF traces of workloads running on Fast Models	109193	Non-Confidential
Introduction to SVE2	102340	Non-Confidential
LISA+ Language for Fast Models Reference Guide	101092	Non-Confidential
Model Debugger for Fast Models User Guide	100968	Non-Confidential
Motherboard Express µATX V2M-P1 Technical Reference Manual	DUI 0447	Non-Confidential
Product Download Hub	–	Non-Confidential
Workload Trace Generation Best Practices	107983	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
AMBA® Low Power Interface Specification	IHI 0068	Non-Confidential
Armv7-M Architecture Reference Manual	DDI 0403	Non-Confidential
Arm® Architecture Reference Manual Armv7-A and Armv7-R edition	DDI 0406	Non-Confidential
Arm® Architecture Reference Manual for A-profile architecture	DDI 0487	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
<i>Arm® Architecture Reference Manual Supplement, The Scalable Vector Extension</i>	DDI 0584	Non-Confidential
<i>Arm® Generic Interrupt Controller Architecture version 2.0 - Architecture Specification</i>	IHI 0048	Non-Confidential
<i>Arm® Power Policy Unit Architecture Specification</i>	DEN 0051	Non-Confidential
<i>Arm® Realm Management Extension (RME) System Architecture</i>	DEN 0129	Non-Confidential

Non-Arm resources	Document ID	Organization
<i>Accellera Systems Initiative</i>	–	http://www.accellera.org
<i>GitHub</i>	–	Github
<i>Intel Download Center, Intel StrataFlash Memory (J3) datasheet</i>	–	Intel
<i>MultiMedia Card Association specification</i>	–	JEDEC
<i>Simple DirectMedia Layer Cross-platform Development Library</i>	–	Simple DirectMedia Layer
<i>Virtual I/O Device (VIRTIO) Version 1.0</i>	–	OASIS Open