



Fast Models

Version 11.24

Reference Guide

Non-Confidential

Copyright © 2017–2023 Arm Limited (or its affiliates).
All rights reserved.

Issue 00

100964_1124_00_en



Fast Models

Reference Guide

Copyright © 2017–2023 Arm Limited (or its affiliates). All rights reserved.

Release Information

Document history

Issue	Date	Confidentiality	Change
1100-00	31 May 2017	Non-Confidential	Update for v11.0. Document numbering scheme has changed.
1101-00	31 August 2017	Non-Confidential	Update for v11.1.
1102-00	17 November 2017	Non-Confidential	Update for v11.2.
1103-00	23 February 2018	Non-Confidential	Update for v11.3.
1104-00	22 June 2018	Non-Confidential	Update for v11.4.
1104-01	17 August 2018	Non-Confidential	Update for v11.4.2.
1105-00	23 November 2018	Non-Confidential	Update for v11.5.
1106-00	27 February 2019	Non-Confidential	Update for v11.6.
1106-01	25 March 2019	Non-Confidential	Update for v11.6.1.
1107-00	17 May 2019	Non-Confidential	Update for v11.7.
1108-00	5 September 2019	Non-Confidential	Update for v11.8.
1109-00	28 November 2019	Non-Confidential	Update for v11.9.
1110-00	12 March 2020	Non-Confidential	Update for v11.10.

Issue	Date	Confidentiality	Change
1111-00	9 June 2020	Non-Confidential	Update for v11.11.
1112-00	22 September 2020	Non-Confidential	Update for v11.12.
1113-00	9 December 2020	Non-Confidential	Update for v11.13.
1114-00	17 March 2021	Non-Confidential	Update for v11.14.
1114-01	30 March 2021	Non-Confidential	Document update 1 for v11.14.
1115-00	29 June 2021	Non-Confidential	Update for v11.15.
1116-00	6 October 2021	Non-Confidential	Update for v11.16.
1117-00	16 February 2022	Non-Confidential	Update for v11.17.
1118-00	15 June 2022	Non-Confidential	Update for v11.18.
1119-00	14 September 2022	Non-Confidential	Update for v11.19.
1120-00	7 December 2022	Non-Confidential	Update for v11.20.
1121-00	22 March 2023	Non-Confidential	Update for v11.21.
1122-00	14 June 2023	Non-Confidential	Update for v11.22.
1123-00	13 September 2023	Non-Confidential	Update for v11.23.
1124-00	6 December 2023	Non-Confidential	Update for v11.24.

Proprietary Notice

This document is protected by copyright and other related rights and the practice 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. 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 implementations infringe any third party patents.

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, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

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.

This document consists solely of commercial items. You shall be responsible for ensuring that any 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 the Agreement shall prevail.

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. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>.

Copyright © 2017–2023 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is Final, that is for a developed product.

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>.

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.

Contents

1. Introduction.....	23
1.1 Conventions.....	23
1.2 Other information.....	24
1.3 Useful resources.....	24
2. About the models.....	27
2.1 Model capabilities.....	27
2.2 Running Arm® Software Test Libraries (STL) on Fast Models.....	27
2.3 Quality level definitions.....	28
2.4 Fast Models accuracy.....	28
2.4.1 Timing accuracy of Fast Models.....	29
2.4.2 Bus traffic in Fast Models.....	29
2.4.3 Instruction prefetch in Fast Models.....	31
2.4.4 Out-of-order execution and write-buffers in Fast Models.....	31
2.4.5 Caches in Fast Models.....	32
2.4.6 Global exclusive monitor in Fast Models.....	33
2.5 Processor implementation.....	33
2.5.1 Mapping PMU events to MTI trace sources.....	33
2.5.2 Caches in PV models.....	35
2.5.3 GICv3 in PV models.....	37
2.5.4 GICv4 in PV models.....	37
2.5.5 CP14 Debug coprocessor.....	38
2.5.6 TLBs in PV models.....	38
2.5.7 Memory access in PV models.....	38
2.5.8 Timing in PV models.....	39
2.5.9 CADI interactions with processor behavior.....	40
2.6 Fast Models CADI implementation.....	41
2.7 CADI sync watchpoints.....	43
2.8 Non-CADI sync watchpoints.....	43
2.8.1 syncLevel definitions.....	44
2.8.2 Controlling and observing the syncLevel.....	45
2.9 SCADI.....	47

2.9.1 Intended uses of CADI and SCADI.....	47
2.9.2 Responsibilities of the SCADI caller.....	48
2.9.3 SCADI interface access.....	48
2.9.4 SCADI semantics.....	50
2.10 User mode networking.....	51
2.11 TAP/TUN networking.....	53
2.11.1 TAP/TUN networking - limitations.....	53
2.11.2 Setting up a network connection for Microsoft Windows.....	53
2.11.3 Configuring the networking environment for Microsoft Windows.....	54
2.11.4 Commands for tap_setup_32.exe or tap_setup_64.exe for Microsoft Windows.....	54
2.11.5 Uninstalling networking for Microsoft Windows.....	55
2.11.6 Setting up a network connection for Red Hat Enterprise Linux.....	55
2.11.7 Setting up a network connection for Ubuntu Linux.....	57
2.11.8 Configuring the networking environment for Linux.....	59
2.11.9 Solutions to networking issues on Linux.....	60
2.11.10 Disabling and re-enabling networking for Linux.....	60
2.11.11 Uninstalling networking for Linux.....	61
2.12 Using parameters to set port values.....	61
2.13 PVBUS C++ transaction and Tx_Result classes.....	62
2.13.1 Class pv::TransactionGenerator.....	62
2.13.2 TransactionGenerator efficiency considerations.....	63
2.13.3 Enum pv::AccessWidth.....	63
2.13.4 Class pv::Transaction.....	63
2.13.5 Class pv::ReadTransaction.....	64
2.13.6 Class pv::WriteTransaction.....	65
2.14 Visualisation library.....	65
2.14.1 LISA visualisation models.....	66
2.14.2 Visualisation library C++ classes.....	66
3. Protocols.....	71
3.1 AMBA-PV protocols.....	71
3.1.1 AMBA-PV protocols - about.....	71
3.1.2 AMBAPV protocol.....	72
3.1.3 AMBAPVACE protocol.....	74
3.1.4 AMBAPVSignal protocol.....	76
3.1.5 AMBAPVSignalState protocol.....	76

3.1.6 AMBAPVValue protocol.....	77
3.1.7 AMBAPVValue64 protocol.....	77
3.1.8 AMBAPVValueState protocol.....	77
3.1.9 AMBAPVValueState64 protocol.....	77
3.2 Clocking protocols.....	78
3.2.1 Clocking protocols - about.....	78
3.2.2 ClockRateControl protocol.....	78
3.2.3 TimerCallback protocol.....	78
3.2.4 TimerCallback64 protocol.....	78
3.2.5 TimerControl protocol.....	79
3.2.6 TimerControl64 protocol.....	79
3.3 Debug interface protocols.....	79
3.3.1 Debug interface protocols - about.....	80
3.3.2 CADIDisassemblerProtocol protocol.....	80
3.3.3 CADIProtocol protocol.....	82
3.4 Peripheral protocols.....	84
3.4.1 AudioControl protocol.....	84
3.4.2 CharacterLCD protocol.....	84
3.4.3 FlashLoaderPort protocol.....	84
3.4.4 GUIPollCallback protocol.....	85
3.4.5 ICS307Configuration protocol.....	85
3.4.6 KeyboardStatus protocol.....	85
3.4.7 LCD protocol.....	86
3.4.8 LCDLayoutInfo protocol.....	86
3.4.9 MMC_Protocol protocol.....	86
3.4.10 MouseStatus protocol.....	87
3.4.11 PL080_DMACE_DmaPortProtocol protocol.....	88
3.4.12 PS2Data protocol.....	89
3.4.13 PVBusSlaveControl protocol.....	89
3.4.14 PVDevice protocol.....	91
3.4.15 PVTransactionMaster protocol.....	92
3.4.16 SerialData protocol.....	92
3.4.17 SMMUv3AEMIdentifyProtocol protocol.....	93
3.4.18 TZFilterControl protocol.....	94
3.4.19 VirtualEthernet protocol.....	94
3.5 Power management protocols.....	95

3.5.1 PChannel protocol.....	95
3.6 Processor protocols.....	96
3.6.1 CoprocBusProtocol protocol.....	96
3.6.2 CounterInterface protocol.....	101
3.6.3 GICv3Comms protocol.....	101
3.6.4 InstructionCount protocol.....	102
3.6.5 v8EmbeddedCrossTrigger_controlprotocol protocol.....	102
3.7 Signaling protocols.....	102
3.7.1 Signaling protocols - about.....	102
3.7.2 Signal protocol.....	103
3.7.3 StateSignal protocol.....	103
3.7.4 Value protocol.....	103
3.7.5 Value_64 protocol.....	103
3.7.6 ValueState protocol.....	103
4. Fast Models components.....	105
4.1 Component differences.....	105
4.2 Bridge components.....	106
4.2.1 AMBAPV2PVBus.....	107
4.2.2 AMBAPVACE2PVBus.....	108
4.2.3 AMBAPVSignal2SGSignal.....	109
4.2.4 AMBAPVSignalState2SGStateSignal.....	109
4.2.5 AMBAPVValue2SGValue.....	109
4.2.6 AMBAPVValue2SGValue64.....	110
4.2.7 AMBAPVValue642SMMUv3AEMIdentify.....	110
4.2.8 AMBAPVValue642VECB.....	110
4.2.9 AMBAPVValueState2SGValueState.....	111
4.2.10 AMBAPVValueState2SGValueState64.....	111
4.2.11 BroadcastSignal2AMBAPVSignal.....	112
4.2.12 Clock2SystemC.....	112
4.2.13 ClockRateConversion.....	112
4.2.14 ClockSignal2SC_ClockSignal.....	113
4.2.15 CoprocBus2SystemC.....	113
4.2.16 CounterInterface2SystemC.....	113
4.2.17 InstructionCount2SystemC.....	114
4.2.18 LCD2SystemC.....	114

4.2.19 PChannel2SystemC.....	115
4.2.20 PVBUS2AMBAPV.....	115
4.2.21 PVBUS2AMBAPVACE.....	117
4.2.22 PVBUSBridge.....	119
4.2.23 SC_ClockSignal2ClockSignal.....	119
4.2.24 SGSignal2AMBAPVSignal.....	120
4.2.25 SGStateSignal2AMBAPVSignalState.....	120
4.2.26 SGValue2AMBAPVValue.....	120
4.2.27 SGValue2AMBAPVValue64.....	121
4.2.28 SGValueState2AMBAPVValueState.....	121
4.2.29 SGValueState2AMBAPVValueState64.....	122
4.2.30 SMMUv3AEMIdentify2AMBAPVValue64.....	122
4.2.31 SystemC2Clock.....	122
4.2.32 SystemC2CoproccBus.....	123
4.2.33 SystemC2CounterInterface.....	123
4.2.34 SystemC2InstructionCount.....	123
4.2.35 SystemC2LCD.....	124
4.2.36 SystemC2PChannel.....	125
4.2.37 SystemC2VirtualEthernet.....	125
4.2.38 SystemC2v7VGICConfig.....	125
4.2.39 VECB2AMBAPVValue64.....	126
4.2.40 VirtualEthernet2SystemC.....	126
4.2.41 v7VGICConfig2SystemC.....	126
4.3 Bus components.....	127
4.3.1 Labeller.....	128
4.3.2 LabellerForDMA330.....	129
4.3.3 LabellerForGPUProtMode.....	130
4.3.4 MSIRewriter.....	131
4.3.5 PASSwitch.....	133
4.3.6 PVBUS4KBTo1KBSplitter.....	133
4.3.7 PVBUSCache.....	134
4.3.8 PVBUSDecoder.....	134
4.3.9 PVBUSExclusiveMonitor.....	135
4.3.10 PVBUSExclusiveSquasher.....	136
4.3.11 PVBUSLogger.....	136
4.3.12 PVBUSMapper.....	137

4.3.13 PVBusMaster.....	138
4.3.14 PVBusModifier.....	138
4.3.15 PVBusRouter.....	139
4.3.16 PVBusSlave.....	139
4.3.17 PVMemoryProtectionEngine.....	140
4.3.18 PVWriteBuffer.....	142
4.3.19 SimplePVBusMaster.....	143
4.3.20 TZSwitch.....	144
4.4 Clocking components.....	145
4.4.1 ClockDivider.....	145
4.4.2 ClockTimer.....	146
4.4.3 ClockTimer64.....	147
4.4.4 ClockTimerThread.....	147
4.4.5 ClockTimerThread64.....	148
4.4.6 MasterClock.....	148
4.5 Core components.....	149
4.5.1 AEMv8RMPCT.....	150
4.5.2 AEMvACT.....	196
4.5.3 ARMAEMv8MCT.....	252
4.5.4 ARMCortexA5CT.....	286
4.5.5 ARMCortexA5MPx1CT.....	292
4.5.6 ARMCortexA7x1CT.....	299
4.5.7 ARMCortexA8CT.....	308
4.5.8 ARMCortexA9MPx1CT.....	315
4.5.9 ARMCortexA9UPCT.....	322
4.5.10 ARMCortexA15x1CT.....	329
4.5.11 ARMCortexA17x1CT.....	338
4.5.12 ARMCortexA32x1CT.....	347
4.5.13 ARMCortexA34x1CT.....	357
4.5.14 ARMCortexA35x1CT.....	367
4.5.15 ARMCortexA53x1CT.....	378
4.5.16 ARMCortexA55CT.....	390
4.5.17 ARMCortexA55CT_CortexA75CT.....	402
4.5.18 ARMCortexA55CT_CortexA76CT.....	421
4.5.19 ARMCortexA55CT_CortexA78CT.....	439
4.5.20 ARMCortexA57x1CT.....	457

4.5.21 ARMCortexA65AECT.....	469
4.5.22 ARMCortexA65AECT_CortexA76AECT.....	481
4.5.23 ARMCortexA65CT.....	499
4.5.24 ARMCortexA72x1CT.....	512
4.5.25 ARMCortexA73x1CT.....	522
4.5.26 ARMCortexA75CT.....	532
4.5.27 ARMCortexA76AECT.....	545
4.5.28 ARMCortexA76CT.....	558
4.5.29 ARMCortexA77CT.....	570
4.5.30 ARMCortexA78AECT.....	583
4.5.31 ARMCortexA78CCT.....	596
4.5.32 ARMCortexA78CT.....	609
4.5.33 ARMCortexA510CT.....	622
4.5.34 ARMCortexA510CT_CortexA710CT.....	638
4.5.35 ARMCortexA510CT_CortexA710CT_CortexX3CT.....	651
4.5.36 ARMCortexA510CT_CortexA715CT_CortexX3CT.....	669
4.5.37 ARMCortexA520CT.....	686
4.5.38 ARMCortexA520CT_CortexA720CT.....	702
4.5.39 ARMCortexA520CT_CortexA720CT_CortexX4CT.....	716
4.5.40 ARMCortexA710CT.....	737
4.5.41 ARMCortexA715CT.....	752
4.5.42 ARMCortexA720CT.....	769
4.5.43 ARMCortexM0CT.....	785
4.5.44 ARMCortexM0PlusCT.....	788
4.5.45 ARMCortexM3CT.....	792
4.5.46 ARMCortexM4CT.....	797
4.5.47 ARMCortexM7CT.....	801
4.5.48 ARMCortexM23CT.....	807
4.5.49 ARMCortexM33CT.....	815
4.5.50 ARMCortexM35PCT.....	825
4.5.51 ARMCortexM52CT.....	835
4.5.52 ARMCortexM55CT.....	838
4.5.53 ARMCortexM85CT.....	850
4.5.54 ARMCortexR4CT.....	862
4.5.55 ARMCortexR5x1CT.....	868
4.5.56 ARMCortexR7x1CT.....	874

4.5.57 ARMCortexR8x1CT.....	880
4.5.58 ARMCortexR52PlusCT.....	888
4.5.59 ARMCortexR52x1CT.....	912
4.5.60 ARMCortexR82CT.....	937
4.5.61 ARMCortexX1CCT.....	951
4.5.62 ARMCortexX1CT.....	963
4.5.63 ARMCortexX2CT.....	976
4.5.64 ARMCortexX3CT.....	990
4.5.65 ARMCortexX4CT.....	1007
4.5.66 ARMNeoverseE1CT.....	1027
4.5.67 ARMNeoverseN1CT.....	1039
4.5.68 ARMNeoverseN2CT.....	1052
4.5.69 ARMNeoverseV1CT.....	1067
4.5.70 ARMNeoverseV2CT.....	1080
4.5.71 ARMSC000CT.....	1095
4.5.72 ARMSC300CT.....	1099
4.6 Media components.....	1103
4.6.1 D71.....	1103
4.6.2 DP500.....	1104
4.6.3 DP500x2.....	1106
4.6.4 DP550.....	1107
4.6.5 DP550x2.....	1109
4.6.6 DP650.....	1111
4.6.7 DP650x2.....	1112
4.6.8 Mali_C5x_streaming_sink.....	1114
4.6.9 Mali_C7x_streaming_sink.....	1115
4.6.10 Mali_C55.....	1115
4.6.11 Mali_C71.....	1119
4.6.12 Mali_C78.....	1120
4.6.13 Mali_Cxx_streaming_camera.....	1121
4.6.14 Mali_G51.....	1122
4.6.15 Mali_G71.....	1124
4.6.16 Mali_G72.....	1125
4.6.17 Mali_G76.....	1127
4.6.18 Mali_G710.....	1129
4.6.19 Mali_G715.....	1131

4.6.20 Mali_G720.....	1134
4.6.21 Mali_T624.....	1135
4.6.22 V61.....	1136
4.6.23 V550.....	1137
4.7 Peripheral components.....	1139
4.7.1 AudioOut_File.....	1139
4.7.2 AudioOut_SDL.....	1140
4.7.3 Base_PowerController.....	1140
4.7.4 DebugAccessPort.....	1142
4.7.5 DebugROM.....	1142
4.7.6 DualClusterSystemConfigurationBlock.....	1142
4.7.7 DummyAPB.....	1145
4.7.8 ElfLoader.....	1146
4.7.9 FlashLoader.....	1147
4.7.10 GICv3CommsLogger.....	1148
4.7.11 GICv3CommsPVBUS.....	1148
4.7.12 GICv3ProtocolChecker.....	1149
4.7.13 GUIPoll.....	1149
4.7.14 HostBridge.....	1150
4.7.15 HostSerialInterface.....	1151
4.7.16 IntelStrataFlashJ3.....	1151
4.7.17 Interrupt_Router.....	1154
4.7.18 LS64TestingFIFO.....	1154
4.7.19 MemoryMappedGenericTimer.....	1155
4.7.20 MemoryMappedGenericWatchdog.....	1156
4.7.21 NonVolatileCounter.....	1157
4.7.22 PCIeATC.....	1158
4.7.23 PPUMultiThreadModeSwitch.....	1159
4.7.24 PS2Keyboard.....	1159
4.7.25 PS2Mouse.....	1160
4.7.26 PVBUSGICv3Comms.....	1160
4.7.27 PVMetaDataController.....	1161
4.7.28 PchannelListener.....	1162
4.7.29 RAMDevice.....	1162
4.7.30 ROM.....	1163
4.7.31 RandomNumberGenerator.....	1164

4.7.32 RealTimeLimiter.....	1164
4.7.33 RealtimeClockTimer.....	1165
4.7.34 RemapDecoder.....	1165
4.7.35 RootKeyStorage.....	1166
4.7.36 SMMUv3TestEngine.....	1167
4.7.37 STLBusGasket.....	1168
4.7.38 SerialCrossover.....	1168
4.7.39 TelnetTerminal.....	1169
4.7.40 UnusedPrimeCell.....	1172
4.7.41 VirtioBlockDevice.....	1173
4.7.42 VirtioBlockDeviceMMIO.....	1173
4.7.43 VirtioNetMMIO.....	1175
4.7.44 VirtioP9Device.....	1177
4.7.45 VirtioPCIBlockDevice.....	1179
4.7.46 VirtioRNG.....	1180
4.7.47 VirtualEthernetCrossover.....	1182
4.7.48 VirtualEthernetHub3.....	1182
4.7.49 VisEventRecorder.....	1183
4.7.50 WarningMemory.....	1184
4.7.51 v8EmbeddedCrossTrigger_Matrix.....	1184
4.8 Scheduler components.....	1184
4.8.1 AsyncSignal.....	1185
4.8.2 SchedulerInterface.....	1185
4.8.3 SchedulerThread.....	1185
4.8.4 SchedulerThreadEvent.....	1186
4.9 Signals components.....	1186
4.9.1 AndGate.....	1186
4.9.2 LabellerMasterIdExtendedIdUserFlag.....	1186
4.9.3 LabellerUserSignals.....	1188
4.9.4 OrGate.....	1188
4.9.5 SGSignalBuffer.....	1189
4.9.6 SignalDriver.....	1189
4.9.7 SignalLogger.....	1190
4.9.8 Value64Logger.....	1190
4.9.9 ValueLogger.....	1191
4.10 SystemIP components.....	1191

4.10.1 AHCI_SATA.....	1192
4.10.2 BP141_TZMA.....	1192
4.10.3 BP147_TZPC.....	1194
4.10.4 CCI400.....	1195
4.10.5 CCI500.....	1197
4.10.6 CCI550.....	1201
4.10.7 CCN502.....	1206
4.10.8 CCN504.....	1209
4.10.9 CCN508.....	1211
4.10.10 CCN512.....	1213
4.10.11 CI700.....	1215
4.10.12 CMN600.....	1221
4.10.13 CMN600AE.....	1228
4.10.14 CMN600CMLHub.....	1236
4.10.15 CMN650.....	1237
4.10.16 CMN650R2.....	1244
4.10.17 CMN700.....	1251
4.10.18 DMA350.....	1262
4.10.19 DMC500.....	1264
4.10.20 DMC520.....	1266
4.10.21 DMC620.....	1267
4.10.22 DMC_400.....	1269
4.10.23 EthosU55.....	1270
4.10.24 EthosU65.....	1271
4.10.25 Firewall.....	1272
4.10.26 GIC500.....	1273
4.10.27 GIC500_ClusterPorts.....	1296
4.10.28 GIC500_Filter.....	1320
4.10.29 GIC600.....	1343
4.10.30 GIC600AE.....	1348
4.10.31 GIC600AE_Filter.....	1353
4.10.32 GIC600_Filter.....	1359
4.10.33 GIC625.....	1364
4.10.34 GIC625_Filter.....	1368
4.10.35 GIC700.....	1373
4.10.36 GIC700_Filter.....	1380

4.10.37 GIC_400.....	1388
4.10.38 GIC_IRI.....	1391
4.10.39 GIC_IRI_Filter.....	1408
4.10.40 GICv3IRI.....	1425
4.10.41 GICv3IRI_Filter.....	1443
4.10.42 ICS307.....	1460
4.10.43 IDAU.....	1462
4.10.44 MMC.....	1463
4.10.45 MMU_400.....	1468
4.10.46 MMU_400_BASE.....	1482
4.10.47 MMU_500.....	1485
4.10.48 MMU_500_BASE.....	1499
4.10.49 MMU_600.....	1502
4.10.50 MMU_700.....	1514
4.10.51 MemoryMappedCounterModule.....	1527
4.10.52 MessageHandlingUnitV2.....	1529
4.10.53 NI700.....	1531
4.10.54 PL011_Uart.....	1534
4.10.55 PL022_SSP.....	1537
4.10.56 PL030_RTC.....	1538
4.10.57 PL031_RTC.....	1539
4.10.58 PL041_AACI.....	1540
4.10.59 PL050_KMI.....	1541
4.10.60 PL061_GPIO.....	1542
4.10.61 PL080_DMAC.....	1543
4.10.62 PL110_CLCD.....	1545
4.10.63 PL111_CLCD.....	1546
4.10.64 PL180_MCI.....	1548
4.10.65 PL192_VIC.....	1549
4.10.66 PL310_L2CC.....	1550
4.10.67 PL330_DMAC.....	1554
4.10.68 PL340_DMC.....	1560
4.10.69 PL350_SMC.....	1562
4.10.70 PL350_SMC_NAND_FLASH.....	1567
4.10.71 PL370_HDLCD.....	1568
4.10.72 PL390_GIC.....	1569

4.10.73 PPUMTWakerequest.....	1575
4.10.74 PPUv0.....	1575
4.10.75 PPUv1.....	1576
4.10.76 PPUv1_Cluster_Wakerequest_Logic.....	1580
4.10.77 SMMUv3AEM.....	1581
4.10.78 SMSC_91C111.....	1606
4.10.79 SP804_Timer.....	1607
4.10.80 SP805_Watchdog.....	1608
4.10.81 SP810_SysCtrl.....	1609
4.10.82 TZC_400.....	1610
4.10.83 TZFilterUnit.....	1612
4.10.84 TZIC.....	1613
4.10.85 VHT_VIOBridge.....	1614
4.10.86 VHT_VSIBridge.....	1614
4.10.87 VHT_VSocket.....	1614
4.10.88 v7_VGIC.....	1615
5. Plug-ins for Fast Models.....	1617
5.1 Loading a plug-in.....	1617
5.1.1 --plugin command-line option.....	1617
5.1.2 scx::scx_load_plugin() method.....	1618
5.1.3 FM_TRACE_PLUGINS environment variable.....	1618
5.2 Customizing a plug-in.....	1619
5.3 ArchMsgTrace.....	1620
5.3.1 ArchMsgTrace - parameters.....	1621
5.4 ASTFplugin.....	1622
5.4.1 ASTFplugin usage notes.....	1623
5.4.2 Additional ASTF support in Fast Models.....	1623
5.4.3 ASTF tools.....	1624
5.4.4 ASTFplugin - parameters.....	1625
5.5 BranchPrediction.....	1625
5.5.1 BranchPrediction parameters.....	1626
5.5.2 BranchPrediction output example.....	1626
5.5.3 Other ways to report branch mispredictions.....	1627
5.6 CADIIPCRemoteConnection.....	1627
5.6.1 CADIIPCRemoteConnection - parameters.....	1628

5.7 CDE.....	1628
5.7.1 CDETester.....	1629
5.7.2 CDETester - parameters.....	1630
5.7.3 CDEConstant.....	1631
5.7.4 CDEConstant - parameters.....	1631
5.7.5 Implementing a CDE plug-in.....	1632
5.7.6 CDE API.....	1634
5.8 Crypto.....	1641
5.8.1 Crypto - parameters.....	1643
5.9 GDBRemoteConnection.....	1644
5.9.1 GDBRemoteConnection parameters.....	1645
5.9.2 GDBRemoteConnection limitations.....	1645
5.10 GenericCounter.....	1646
5.10.1 GenericCounter - parameters.....	1646
5.11 GenericTrace.....	1647
5.11.1 Mapping between SYSREG_UPDATE trace sources and SPSR registers.....	1648
5.11.2 GenericTrace - parameters.....	1648
5.12 libete-plugin.....	1649
5.12.1 libete-plugin - parameters.....	1649
5.13 ListTraceSources.....	1651
5.13.1 ListTraceSources - parameters.....	1652
5.14 PipelineModel.....	1652
5.14.1 PipelineModel parameters.....	1654
5.14.2 PipelineModel example.....	1654
5.14.3 Naming the plug-in instance.....	1655
5.14.4 Example command lines.....	1656
5.14.5 PipelineModel output.....	1658
5.15 RunTimeParameterTest.....	1658
5.16 ScalableVectorExtension.....	1659
5.16.1 ScalableVectorExtension - parameters.....	1659
5.17 Sidechannel.....	1662
5.17.1 Sidechannel - parameters.....	1663
5.18 TarmacText.....	1663
5.18.1 TarmacText - parameters.....	1664
5.19 TarmacTrace.....	1664
5.19.1 TarmacTrace - parameters.....	1664

5.19.2 TarmacTrace file format.....	1667
5.19.3 Tarmac Trace output example.....	1668
5.19.4 Instruction trace.....	1670
5.19.5 Program flow trace.....	1671
5.19.6 Register trace.....	1673
5.19.7 Cache maintenance trace.....	1674
5.19.8 Cache content trace.....	1675
5.19.9 Translation table walk trace.....	1676
5.19.10 TLB trace.....	1677
5.19.11 Event trace.....	1679
5.19.12 Processor memory access trace.....	1680
5.19.13 Processor memory update trace.....	1681
5.19.14 Memory bus trace.....	1683
5.20 ToggleMTIPlugin.....	1685
5.20.1 How to use ToggleMTIPlugin.....	1685
5.20.2 ToggleMTIPlugin - parameters.....	1686
6. Fast Models examples.....	1688
6.1 CADI examples.....	1688
6.2 LISA examples.....	1689
6.3 Build and run an FVP example.....	1690
6.4 LISAPlus examples.....	1694
6.5 MTI examples.....	1694
6.6 SystemCExport examples.....	1695
6.7 Build and run an EVS platform example.....	1696
7. Base Platform.....	1698
7.1 About the Base Platform.....	1698
7.2 Base Platform RevC.....	1699
7.2.1 Base Platform RevC block diagram.....	1700
7.2.2 BasePlatformPCIRevC component.....	1700
7.2.3 Base Platform RevC PCIeRootComplex.....	1703
7.2.4 Base Platform RevC SMMUv3AEM.....	1704
7.2.5 Base Platform RevC PCIe device assignment.....	1705
7.2.6 Base Platform RevC legacy PCI interrupts.....	1706
7.2.7 Base Platform RevC MSI-X.....	1706
7.3 Base Platform startup configuration for v9 cores.....	1706

7.4 Base - memory.....	1707
7.4.1 Base - secure memory.....	1707
7.4.2 Base Platform memory map.....	1708
7.4.3 BaseR Platform memory map.....	1711
7.4.4 Base - DRAM.....	1712
7.5 Base - interrupt assignments.....	1712
7.6 Base - clocks.....	1714
7.7 Base - parameters.....	1715
7.8 Base - components.....	1716
7.8.1 Base - components - about.....	1716
7.8.2 Base - Base_PowerController component.....	1716
7.8.3 Base - DebugAccessPort component.....	1721
7.8.4 Simulator visualization - parameters.....	1722
7.8.5 Base - VE_SysRegs component.....	1722
7.9 Base - VE compatibility.....	1724
7.9.1 Base - VE compatibility - GICv2.....	1724
7.9.2 Base - VE compatibility - GICv3.....	1725
7.9.3 Base - VE compatibility - system global counter.....	1725
7.9.4 Base - VE compatibility - disable security.....	1725
7.10 Base - unsupported VE features.....	1726
7.10.1 Base - unsupported VE features - memory aliasing at 0x08_00000000.....	1726
7.10.2 Base - unsupported VE features - boot ROM alias at 0x00_0800_0000.....	1726
7.10.3 Base - unsupported VE features - change of older parameters.....	1726
8. Microcontroller Prototyping System 2.....	1728
8.1 MPS2 - about.....	1728
8.2 MPS2 platform types.....	1728
8.3 MPS2 - memory maps.....	1729
8.3.1 MPS2 - memory map for models without the Arm®v8-M additions.....	1729
8.3.2 MPS2 - memory map for models with the Arm®v8-M additions.....	1730
8.4 MPS2 - interrupt assignments.....	1737
8.5 MPS2 - differences between models and hardware.....	1738
9. Versatile Express Model.....	1740
9.1 VE - about.....	1740
9.2 VE memory map for Cortex®-A series.....	1742

9.3 VE memory map for Cortex®-R series.....	1744
9.4 VE - interrupt assignments for Cortex®-A series.....	1745
9.5 VE - interrupt assignments for Cortex®-R series.....	1746
9.6 VE parameters.....	1747
9.6.1 VE instantiation parameters.....	1747
9.6.2 VE secure memory parameters.....	1747
9.6.3 VE switch S6.....	1747
9.7 VE - components.....	1748
9.7.1 VEVisualisation component.....	1749
9.7.2 VE_SysRegs component.....	1752
9.8 Differences between the VE hardware and the system models.....	1754
9.8.1 Memory map.....	1755
9.8.2 Memory aliasing.....	1755
9.8.3 VE hardware features absent.....	1755
9.8.4 VE hardware features different.....	1756
9.8.5 Timing considerations for the VE FVPs.....	1756

1. Introduction

This document provides a reference for all Fast Models components, plug-ins, and examples included in the Fast Models Portfolio.

1.1 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.

Convention	Use
<i>italic</i>	Citations.
bold	Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace <u>underline</u>	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: <pre>MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2></pre>
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 .



Caution

Recommendations. Not following these recommendations might lead to system failure or damage.



Warning

Requirements for the system. Not following these requirements might result in system failure or damage.



Requirements for the system. Not following these requirements will result in system failure or damage.



An important piece of information that 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.

1.2 Other information

See the Arm® website for other relevant information.

- [Arm® Developer](#).
- [Arm® Documentation](#).
- [Technical Support](#).
- [Arm® Glossary](#).

1.3 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
AMBA-PV Extensions to TLM User Guide	100962	Non-Confidential
AN400 Arm® Cortex®-M7 SMM on V2M-MPS2 Application Note 400	DAI 0400	Non-Confidential
AN521 Example SSE-200 Subsystem for MPS2+ Application Note	DAI 0521	Non-Confidential
A-Profile Architecture	-	Non-Confidential
Arm® Architecture Models	-	Non-Confidential
Arm® Corstone™ SSE-300 Example Subsystem Technical Reference Manual	101773	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
Cortex®-M23 processor Arm®v8-M IoT Kit User Guide	ECM 0635473	Non-Confidential
Cortex®-M33 processor Arm®v8-M IoT Kit FVP User Guide	ECM 0601256	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
Fast Models Fixed Virtual Platforms (FVP) Reference Guide	100966	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
Arm® Architecture Reference Manual Arm®v7-A and Arm®v7-R edition	DDI 0406	Non-Confidential
Arm® Architecture Reference Manual for A-profile architecture	DDI 0487	Non-Confidential
Arm® Architecture Reference Manual Supplement, The Scalable Vector Extension	DDI 0584	Non-Confidential
Arm® Generic Interrupt Controller Architecture version 2.0 - Architecture Specification	IHI 0048	Non-Confidential
Arm® Power Policy Unit Architecture Specification	DEN 0051	Non-Confidential
Arm® Realm Management Extension (RME) System Architecture	DEN 0129	Non-Confidential
Arm®v7-MArchitecture Reference Manual	DDI 0403	Non-Confidential

Non-Arm® resources	Document ID	Organization
Accellera Systems Initiative (ASI)	-	Accellera Systems Initiative
Intel Download Center, Intel StrataFlash Memory (J3) datasheet	-	Intel
MultiMedia Card Association specification	-	JEDEC
Simple DirectMedia Layer Cross-platform Development Library	-	Simple DirectMedia Layer
Virtual I/O Device (VIRTIO) Version 1.0	-	OASIS Open

**Note**

Arm tests its PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its documents when used with any other PDF reader.

Adobe PDF reader products can be downloaded at <http://www.adobe.com>.

2. About the models

Programmer's 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.

2.1 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](#) on page 31

2.2 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

[Software Test Libraries](#)

2.3 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 2-1: 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.

2.4 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.

2.4.1 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.



If timing annotation is enabled, the cycle count and instruction count can 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.

2.4.2 Bus traffic in Fast Models

PVBus can simulate the behavior of individual bus transactions passing through a hierarchy of bus fabric, but it employs several techniques to optimize this process.

- PVBus generally decodes the path between a bus master and the 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”, that 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.

Related information

[Instruction prefetch in Fast Models](#) on page 30

2.4.3 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 block size 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.

Related information

[Bus traffic in Fast Models](#) on page 29

2.4.4 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.

2.4.5 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.

2.4.6 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.

2.5 Processor implementation

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

2.5.1 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 2-2: 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. ¹

Table 2-3: PMUv2 event mappings

PMUv2 event	MTI source	Notes
Data memory access (0x13)	-	Not modeled. ¹
Level 1 instruction cache access (0x14)	CACHE_READ_HIT, CACHE_READ_MISS	-
Level 1 data cache write-back (0x15)	-	Not modeled. ¹
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. ¹
Bus access (0x19)	CORE_LOADS, CORE_STORES, MEMTAG_LOADS, MEMTAG_STORES	-
Memory error (0x1A)	-	Not modeled. ¹
Instruction speculatively executed (0x1B)	-	Not modeled. ¹

¹ Not modeled means that the counter value is zero and does not increment.

PMUv2 event	MTI source	Notes
Write to TTBR (0x1C)	TTBR_WRITE	-
Bus cycle (0x1D)	INST	This value is an approximation.

Related information

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

2.5.2 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.

2.5.2.1 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.

2.5.2.2 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.

2.5.3 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) and the *Interrupt Redistribution Infrastructure* (IRI). You can configure all Arm®v8-A cores to include a GICv3 interface. You can integrate a separate GICv3IRI 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 GICv3IRI component and 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.

Related information

[GICv3IRI](#) on page 1425

2.5.4 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 `GICv3IRI` or `GICv3IRI_Filter` component.

In addition to requiring the presence of an ITS that is configured as shown in [4.10.40 GICv3IRI](#) on page 1425, GICv4 requires you to enable the virtual LPIs feature and to configure a virtual PE table using the parameters shown in the following example:

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

2.5.5 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.

2.5.6 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` parameter 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.

2.5.7 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.

2.5.7.1 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.

2.5.7.2 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.

2.5.8 Timing in PV models

Programmer's 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.

2.5.9 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 are 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.

2.6 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 2-4: 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 2-5: 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 ²	-
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 ²	-
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	-

² Not permitted while target is running.

2.7 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.

2.8 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.

2.8.1 syncLevel definitions

Definitions for 0 - OFF, 1 - SYNC_STATE, 2 - POST_INSN_IO, and 3 - POST_INSN_ALL.

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.

2.8.2 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 `*Register` and `*unregister` 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 the intended 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.

2.9 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.

2.9.1 Intended uses of CADI and SCADI

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

CADI

is 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 functions of CADI, 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()` 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.

2.9.2 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.

2.9.3 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.

2.9.3.1 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.

2.9.3.2 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.

2.9.3.3 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.

2.9.3.4 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()`.

Retrieving a SCADI interface pointer of a specific component

```
#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));
}

```

2.9.4 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 not supported, and return `CADI_STATUS_CmdNotSupported`.

2.10 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.



Note

- 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](#) on page 1149

2.11 TAP/TUN networking

This section describes TAP/TUN networking.

2.11.1 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.

2.11.2 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 Fast Model Portfolio, default `C:\Program Files\ARM\FastModelPortfolio_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**.

2.11.3 Configuring the networking environment for Microsoft Windows

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 and build a model, or use Model Shell or Model Debugger to start a prebuilt model.

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. Optional: 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.

2.11.4 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.

2.11.5 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\FastModelPortfolio_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**.

2.11.6 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
```

2.11.7 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 network configuration can cause the computer to fail to connect to the network. ARM recommends performing these steps on a machine that you have physical access to.

Procedure

1.



Edit the interfaces file with root privileges.

Add `eth0` to `/etc/network/interfaces` if it is not there.

This step stops network-manager from managing `eth0`. It can result in network-manager indicating there is no network connection even if there is.

Option
For an interface using DHCP**Description**

Add:

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

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 makes sure 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.

Run:

```
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 an 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 that is 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 the following:

```
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
```

7. Restart network services with one of the following options:

- Run the following commands:

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



`armbr0` must be explicitly started.

- Restart the computer.

Both choices disconnect and reconnect all network interfaces.

2.11.8 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>
smc_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 User Guide](#)

2.11.9 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: open `/etc/udev/rules.d/50-udev.rules` as root, and find the line:

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

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

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

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

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



Note

Refer to the vendor documentation manual.

2.11.10 Disabling and re-enabling networking for Linux

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

About this task



Caution

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

Procedure

- 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
```

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

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

2.11.11 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.
- `/sbin/brctl`.

You must remove any symlinks that you have created. Removing `brctl` is optional.

2.12 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.

2.13 PVBUS C++ transaction and Tx_Result classes

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

2.13.1 Class `pv::TransactionGenerator`

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

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

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

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

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

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

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

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

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

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

    // Generate read transactions.
```

```

bool read8(bus_addr_t, uint8_t *data);
bool read16(bus_addr_t, uint16_t *data);
bool read32(bus_addr_t, uint32_t *data);
bool read64(bus_addr_t, uint64_t *data);

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

```

2.13.2 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, ARM recommends 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 DMA example in the examples directory of the System Canvas Model Library distribution.

2.13.3 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`

2.13.4 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();

};
```

2.13.5 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();
};
```

2.13.6 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 will be 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 will be 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 will be 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 will be extended as appropriate.
     */
    uint8_t getData8() const;

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

2.14 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 provide 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 85

[LISA visualisation models](#) on page 66

[Visualisation library C++ classes](#) on page 66

2.14.1 LISA visualisation models

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

Each example platform model, such as the VE, contains its own LISA visualisation model. You can use the model as the basis for your own visualization containing components, such as a 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 1740

[Fast Models User Guide](#)

2.14.2 Visualisation library C++ classes

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

2.14.2.1 C++ classes inclusion

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

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

2.14.2.2 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 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.

2.14.2.3 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(int x, int y, unsigned int width, unsigned int height)

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

2.14.2.4 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)

Sets 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(int code)

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

2.14.2.5 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`.

2.14.2.6 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 stick 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]`.

2.14.2.7 Struct VisPixelFormat

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

The members are:

uint32_t rbits, 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 $(R \ll \text{format.roff}) + (G \ll \text{format.goff}) + (B \ll \text{format.boff})$, where (R,G,B) represents the values of the color channels for the pixel, containing values from 0 up to $(1 \ll \text{format.rbits})$, $(1 \ll \text{format.gbits})$, $(1 \ll \text{format.bbits})$.

3. Protocols

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

3.1 AMBA-PV protocols

This section describes the AMBA-PV protocols.

3.1.1 AMBA-PV protocols - about

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

[Bridge components](#) on page 106

[AMBA-PV Extensions to TLM 2.0 Developer Guide](#)

3.1.2 AMBAPV protocol

The AMBAPV protocol defines behaviors for single read and single write transactions. This covers Arm® AMBA® 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.

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
```

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.

3.1.3 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 operations.
- 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::sc_time &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::sc_time &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 and 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

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

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

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, for example 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.

3.1.4 AMBAPVSignal protocol

This protocol defines a single behavior to permit masters to change the state of signals such as interrupt. Strictly speaking, 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.

3.1.5 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. Strictly speaking 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.

3.1.6 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.

3.1.7 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.

3.1.8 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.

3.1.9 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.

3.2 Clocking protocols

This section describes the clocking protocols.

3.2.1 Clocking protocols - about

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 144

3.2.2 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.

$$\text{clk_out_freq} = \text{clk_in_freq} * \text{mul} / \text{div}$$

3.2.3 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 is called again or 0 to make the timer one-shot.

3.2.4 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 is called again or 0 to make the timer one-shot.

3.2.5 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 countdown 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.

3.2.6 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 countdown 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.

3.3 Debug interface protocols

This section describes the debug interface protocols.

3.3.1 Debug interface protocols - about

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) interface level, where the component code takes responsibility for the implementation of these interfaces.

Related information

[Component Architecture Debug Interface User Guide v2.0](#)

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

3.3.2 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 81

[Component Architecture Debug Interface User Guide Version 2.0](#)

3.3.3 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.

3.4 Peripheral protocols

This section describes the peripheral protocols.

3.4.1 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.

3.4.2 CharacterLCD protocol

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

setLayoutInfo

sets the width and height of the touchscreen.

draw

sets the character information.

3.4.3 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.

3.4.4 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.

3.4.5 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.

3.4.6 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 `components/KeyCode.h` for a list of `ATKeyCode` code values.

The protocol behaviors are:

keyDown(ATKeyCode code) : void

This is sent when a key on the host keyboard is pressed.

keyUp(ATKeyCode code) : void

This is sent when a key on the host keyboard is released.

Related information

[VEVisualisation - ports](#) on page 1750

3.4.7 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 1750

3.4.8 LCDLayoutInfo protocol

This protocol has the behavior `setLayoutInfo`.

setLayoutInfo

sets the width and height of the touchscreen.

3.4.9 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.

3.4.10 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 1750

3.4.11 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.

3.4.12 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.

3.4.13 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 PVBusSlave 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 PVBusSlave 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 PVBUSSlave 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, PVBUSSlave 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 PVBUSSlave 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, PVBUSSlave 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, used when Timing Annotation is enabled.

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.

3.4.14 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 PVBUSlave 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::ReadTransaction tx) : pv::Tx_Result
```

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

slave behavior debugWrite(pv::WriteTransaction tx) : 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 62

3.4.15 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;
}
```

3.4.16 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 3-1: 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 3-2: 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 3-3: 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 3-4: Bits for signalsGet()

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

3.4.17 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.

3.4.18 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.

3.4.19 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`.

3.5 Power management protocols

This section describes the power management protocols.

3.5.1 PChannel protocol

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

The behaviors of the protocol are:

active (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 `active(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)`.



Note

- For a LISA+ example that demonstrates how to use PChannel, see `$PVLIB_HOME/examples/LISA/VP_PChannel/`.
- For SystemC test code that demonstrates how to use PChannel, see `$PVLIB_HOME/Platforms/SystemCExport/EVS_Components/EVS_PChannel/`.

3.6 Processor protocols

This section describes the processor protocols.

3.6.1 CoprocBusProtocol protocol

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

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 `addCoprocessor()` 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
    TestValCoproprocessor test_cp;
}

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

```

3.6.1.1 Coprocessor behaviors

A coprocessor can call the following master behaviors:

addCoproprocessor(Coproprocessor*, int num) : void

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

removeCoproprocessor(Coproprocessor*, 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.

3.6.1.2 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 [3.6.1.3 CoprocState values](#) on page 101.

3.6.1.2.1 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 [3.6.1.3 CoprocState values](#) on page 101.

3.6.1.2.2 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 [3.6.1.3 CoprocState values](#) on page 101.

3.6.1.2.3 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 [3.6.1.3 CoprocState values](#) on page 101.

3.6.1.2.4 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 [3.6.1.3 CoprocState values](#) on page 101.

3.6.1.2.5 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 [3.6.1.3 CoprocState values](#) on page 101.

3.6.1.2.6 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 [3.6.1.3 CoprocState values](#) on page 101.

3.6.1.2.7 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 [3.6.1.3 CoprocState values](#) on page 101.

3.6.1.3 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 3-5: CoprocState values

Value	State label	Description
0x0	<code>CoprocOk</code>	Complete/Ok.
0x1	<code>CoprocUndef</code>	Undefined operation.
0x2	<code>CoprocAbort</code>	Data abort.
0x4	<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.
0x5	<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.
0x12	<code>CoprocNop</code>	Treat as a NOP.

3.6.2 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.

3.6.3 GICv3Comms protocol

This protocol connects the `gicv3_redistributor_s` port on components with GICv3 Core Interfaces to platform level GICv3 components.

This semi-opaque protocol is exportable across a SystemC interface using the bridges `GICv3CommsPVBUS` and `PVBUSGICv3Comms`.

It is a LISA+ protocol.

3.6.4 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 3-6: Run state values

Value	State label	Description
0x0	UNKNOWN	Run status unknown, that is, simulation has not started
0x1	RUNNING	Processor running, is not idle and is executing instructions
0x2	HALTED	External halt signal asserted
0x3	STANDBY_WFE	Last instruction executed was WFE and standby mode has been entered
0x4	STANDBY_WFI	Last instruction executed was WFI and standby mode has been entered
0x5	IN_RESET	External reset signal asserted
0x6	DORMANT	Partial processor power down
0x7	SHUTDOWN	Complete processor power down

3.6.5 v8EmbeddedCrossTrigger_controlprotocol protocol

This protocol connects the *Cross Trigger Interface* (CTI) on ARMv8 processor components to platform level *Cross Trigger Matrix* (CTM) components.

This opaque protocol is not exportable across a SystemC interface.

3.7 Signaling protocols

This section describes the signaling protocols.

3.7.1 Signaling protocols - about

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.

3.7.2 Signal protocol

The Signal protocol has the setValue behavior.

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

Legal values for sg::Signal::State are:

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

3.7.3 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.

3.7.4 Value protocol

The Value protocol has the setValue behavior.

setValue(uint32_t value)
indicates a state change.

3.7.5 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.

3.7.6 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.

4. 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 lists the ports and parameters.

4.1 Component differences

This topic lists the new and changed components in this release.

Differences between 11.23.9 and 11.24.4

Table 4-1: Components added

Component	Quality level
ARMCortexM52CT	Preliminary support
DMA350	Preliminary support
LS64TestingFIFO	N/A
Mali_C55	Preliminary support
Mali_C5x_streaming_sink	Full support

Table 4-2: 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	No	No	Yes
ARMCortexA510CT_CortexA710CT	No	No	No	Yes
ARMCortexA510CT_CortexA710CT_CortexX3CT	No	No	No	Yes
ARMCortexA510CT_CortexA715CT_CortexX3CT	No	No	Yes	Yes
ARMCortexA520CT	No	No	No	Yes
ARMCortexA520CT_CortexA720CT	No	No	No	Yes
ARMCortexA520CT_CortexA720CT_CortexX4CT	No	No	No	Yes
ARMCortexA55CT	No	No	No	Yes
ARMCortexA55CT_CortexA75CT	No	No	No	Yes
ARMCortexA55CT_CortexA76CT	No	No	No	Yes
ARMCortexA55CT_CortexA78CT	No	No	No	Yes
ARMCortexA65AECT	No	No	No	Yes
ARMCortexA65AECT_CortexA76AECT	No	No	No	Yes
ARMCortexA65CT	No	No	No	Yes
ARMCortexA715CT	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?
ARMCortexA720CT	No	No	No	Yes
ARMCortexA76AECT	No	No	No	Yes
ARMCortexA76CT	No	No	No	Yes
ARMCortexA77CT	No	No	No	Yes
ARMCortexA78AECT	No	No	No	Yes
ARMCortexA78CCT	No	No	No	Yes
ARMCortexA78CT	No	No	No	Yes
ARMCortexM23CT	No	No	Yes	Yes
ARMCortexR52PlusCT	No	No	No	Yes
ARMCortexR52x1CT	No	No	No	Yes
ARMCortexR82CT	No	No	No	Yes
ARMCortexX1CCT	No	No	No	Yes
ARMCortexX1CT	No	No	No	Yes
ARMCortexX3CT	No	No	No	Yes
ARMCortexX4CT	No	No	No	Yes
ARMNeoverseE1CT	No	No	No	Yes
ARMNeoverseN1CT	No	No	No	Yes
ARMNeoverseV1CT	No	No	No	Yes
ARMNeoverseV2CT	No	No	No	Yes
CMN700	No	No	No	Yes
GIC700	No	No	No	Yes
GIC700_Filter	No	No	No	Yes
GIC_IRI	No	No	No	Yes
GIC_IRI_Filter	No	No	No	Yes
MMU_600	No	No	No	Yes
MMU_700	No	No	No	Yes
PVBusSlave	No	No	No	Yes
PVMetaDataController	No	No	No	Yes
RAMDevice	No	No	No	Yes
SMMUv3AEM	No	No	No	Yes

4.2 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`. On Linux, use the `$MAXCORE_HOME` environment variable instead.

For more information about ASI TLM 2.0, see the Accellera documentation that is provided with the kit.

Related information

[AMBA-PV protocols - about](#) on page 71

[AMBA-PV Extensions to TLM 2.0 Developer Guide](#)

[Fast Models User Guide, SystemC Export with Multiple Instantiation](#)

[Accellera Systems Initiative](#)

4.2.1 AMBAPV2PVBUS

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

AMBAPV2PVBUS contains the following CADI targets:

- AMBAPV2PVBUS

AMBAPV2PVBUS contains the following MTI components:

- [PVBUSMaster](#)

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.

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 4-3: Ports

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

Parameters for AMBAPV2PVBUS

base_addr

Base address.

Type: `int`. Default value: `0x0`.

shareable

Shareable default.

Type: `bool`. Default value: `0x1`.

4.2.2 AMBAPVACE2PVBUS

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

AMBAPVACE2PVBUS contains the following CADI targets:

- AMBAPVACE2PVBUS

AMBAPVACE2PVBUS contains the following MTI components:

- [PVBUSMaster](#)

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 4-4: Ports

Name	Protocol	Type	Description
amba_pv_ace_s	AMBAPVACE	Slave	-
pvbust_m	PVBUS	Master	-

4.2.3 AMBAPVSignal2SGSignal

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



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

Ports for AMBAPVSignal2SGSignal

Table 4-5: 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.

4.2.4 AMBAPVSignalState2SGStateSignal

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



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

Ports for AMBAPVSignalState2SGStateSignal

Table 4-6: Ports

Name	Protocol	Type	Description
amba_pv_signal_s	AMBAPVSignalState	Slave	-
sg_signal_m	StateSignal	Master	-

4.2.5 AMBAPVValue2SGValue

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



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

Ports for AMBAPVValue2SGValue

Table 4-7: Ports

Name	Protocol	Type	Description
amba_pv_value_s	AMBAPVValue	Slave	-
sg_value_m	Value	Master	-

4.2.6 AMBAPVValue2SGValue64

AMBA-PV Value64 to SystemGenerator Value_64 protocol converter. This model is written in LISA+.



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

Ports for AMBAPVValue2SGValue64

Table 4-8: Ports

Name	Protocol	Type	Description
amba_pv_value_s	AMBAPVValue64	Slave	-
sg_value_m	Value_64	Master	-

4.2.7 AMBAPVValue642SMMUv3AEMIdentify

AMBA-PV Value64 to SMMUv3AEMIdentify protocol converter. This model is written in LISA+.

Ports for AMBAPVValue642SMMUv3AEMIdentify

Table 4-9: Ports

Name	Protocol	Type	Description
identify	SMMUv3AEMIdentifyProtocol	Master	-
identify_reply	AMBAPVValue64	Master	-
identify_request	AMBAPVValue64	Slave	-

4.2.8 AMBAPVValue642VECB

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

AMBAPVValue642VECB contains the following CADI targets:

- AMBAPVValue642VECB

Ports for AMBAPVValue64VECB

Table 4-10: Ports

Name	Protocol	Type	Description
amba_pv_ctrl_s	AMBAPVValue	Slave	-
amba_pv_data_s	AMBAPVValue64	Slave	-
vecb_m	VECBProtocol	Master	-

4.2.9 AMBAPVValueState2SGValueState

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



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

Ports for AMBAPVValueState2SGValueState

Table 4-11: Ports

Name	Protocol	Type	Description
amba_pv_value_s	AMBAPVValueState	Slave	-
sg_value_m	ValueState	Master	-

4.2.10 AMBAPVValueState2SGValueState64

AMBA-PV ValueState64 to SystemGenerator ValueState_64 protocol converter. This model is written in LISA+.



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

Ports for AMBAPVValueState2SGValueState64

Table 4-12: Ports

Name	Protocol	Type	Description
amba_pv_value_s	AMBAPVValueState64	Slave	-
sg_value_m	ValueState_64	Master	-

4.2.11 BroadcastSignal2AMBAPVSignal

Broadcast signal to AMBAPVSignal converter. This model is written in LISA+.

BroadcastSignal2AMBAPVSignal contains the following CADI targets:

- BroadcastSignal2AMBAPVSignal

Ports for BroadcastSignal2AMBAPVSignal

Table 4-13: Ports

Name	Protocol	Type	Description
amba_pv_signal_m	AMBAPVSignal	Master	-
amba_pv_signal_s	AMBAPVSignal	Slave	-
b_signal	Signal	Broadcast	-

4.2.12 Clock2SystemC

Clock to SystemC Converter. This model is written in LISA+.

Clock2SystemC contains the following CADI targets:

- Clock2SystemC

Ports for Clock2SystemC

Table 4-14: 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	-

4.2.13 ClockRateConversion

ClockRateControl to rate in Hz (Value_64) Converter. This model is written in LISA+.

ClockRateConversion contains the following CADI targets:

- ClockDivider
- ClockRateConversion

ClockRateConversion contains the following MTI components:

- [ClockDivider](#)

Ports for ClockRateConversion

Table 4-15: Ports

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

4.2.14 ClockSignal2SC_ClockSignal

ClockSignal to SystemC ClockSignal converter. This model is written in LISA+.

ClockSignal2SC_ClockSignal contains the following CADI targets:

- ClockSignal2SC_ClockSignal

Ports for ClockSignal2SC_ClockSignal

Table 4-16: Ports

Name	Protocol	Type	Description
clk_out	ClockSignal	Slave	-
sc_clk_out	SC_ClockSignal	Master	-

4.2.15 CoprocBus2SystemC

CoprocBusProtocol to SystemCCoprocBusProtocol converter. This model is written in LISA+.

CoprocBus2SystemC contains the following CADI targets:

- CoprocBus2SystemC

Ports for CoprocBus2SystemC

Table 4-17: Ports

Name	Protocol	Type	Description
coproc_bus_s	CoprocBusProtocol	Slave	-
sc_coproc_bus_m	SystemCCoprocBusProtocol	Master	-

4.2.16 CounterInterface2SystemC

CounterInterface to SystemC Converter. This model is written in LISA+.

CounterInterface2SystemC contains the following CADI targets:

- CounterInterface2SystemC

Ports for CounterInterface2SystemC

Table 4-18: 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	-

4.2.17 InstructionCount2SystemC

InstructionCount to SystemC Converter. This model is written in LISA+.

InstructionCount2SystemC contains the following CADI targets:

- InstructionCount2SystemC



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

Ports for InstructionCount2SystemC

Table 4-19: Ports

Name	Protocol	Type	Description
inst_count	AMBAPVValueState64	Slave	-
run_state	AMBAPVValueState	Slave	-
ticks	InstructionCount	Slave	-

4.2.18 LCD2SystemC

Converts LCD protocol to SystemC. This model is written in LISA+.

LCD2SystemC contains the following CADI targets:

- LCD2SystemC

Ports for LCD2SystemC

Table 4-20: 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	-

4.2.19 PChannel2SystemC

PChannel to SystemC Converter. This model is written in LISA+.

PChannel2SystemC contains the following CADI targets:

- PChannel2SystemC

Ports for PChannel2SystemC

Table 4-21: Ports

Name	Protocol	Type	Description
pchannel	PChannel	Slave	-
sc_pchannel	SystemCPChannel	Master	-

4.2.20 PVBus2AMBAPV

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

PVBus2AMBAPV contains the following CADI targets:

- PVBus2AMBAPV

PVBus2AMBAPV contains the following MTI components:

- PVBusBridge

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 4-22: Ports

Name	Protocol	Type	Description
<code>amba_pv_m</code>	AMBAPV	Master	-
<code>pvbuss_s</code>	PVBUS	Slave	-

Parameters for PVBUS2AMBAPV

`counters-file-name`

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

Type: `string`. Default value: `""`.

dmi-container-type

Type of the DMI cache. Allowed values: `TZAttr` and `FullAttr`.

Type: `string`. Default value: `"FullAttr"`.

dump-dmi-cache

Dumps the content of the DMI cache into a file.

Type: `bool`. Default value: `0x0`.

dump-dmi-file-name

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

Type: `string`. Default value: `""`.

force-dmi-size

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

Type: `bool`. Default value: `0x1`.

min-range-to-cache

Min DMI range size to cache in the bridge.

Type: `int`. Default value: `0x10000`.

size

Maximum size of memory region.

Type: `int`. Default value: `0x100000000000000`.

4.2.21 PVBus2AMBAPVACE

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

PVBus2AMBAPVACE contains the following CADI targets:

- PVBus2AMBAPVACE

PVBus2AMBAPVACE contains the following MTI components:

- [PVBus2AMBAPVACE](#)
- [PVBusBridge](#)
- [PVBusMapper](#)

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 be able 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 PVBUS2AMBAPVACE

Table 4-23: Ports

Name	Protocol	Type	Description
<code>amba_pv_ace_m</code>	AMBAPVACE	Master	-
<code>pvbust_over_tlm_control</code>	<code>PVBusOverTLMControl</code>	Slave	-
<code>pvbust_s</code>	<code>PVBus</code>	Slave	-

Parameters for PVBUS2AMBAPVACE

counters-file-name

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

Type: `string`. Default value: `""`.

dmi-cache-name

DEPRECATED: This parameter will be ignored. Name of the DMI cache. Useful for multiple bridges to share the same cache.

Type: `string`. Default value: `""`.

dmi-container-type

Type of the DMI cache. Allowed values: `TZAttr` and `FullAttr`. When there are caches downstream of this bridge and `cache-state-modelled=1` and `route-tlm=1` use only `FullAttr`.

Type: `string`. Default value: `"FullAttr"`.

dump-dmi-cache

Dumps the content of the DMI cache into a file.

Type: `bool`. Default value: `0x0`.

dump-dmi-file-name

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

Type: `string`. Default value: `""`.

force-dmi-size

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

Type: `bool`. Default value: `0x1`.

min-range-to-cache

Min DMI range size to cache in the bridge.

Type: `int`. Default value: `0x10000`.

route-tlm

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.

Type: `bool`. Default value: `0x1`.

route-tlm-filter

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}].

Type: string. Default value: "".

set-ace-lite

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

Type: bool. Default value: 0x0.

size

Maximum size of memory region, i.e. the first unsupported address.

Type: int. Default value: 0x1000000000000.

4.2.22 PVBusBridge

A PVBusBridge bridges incoming transactions to a PVDevice port. This model is written in C++.

PVBusBridge contains the following MTI components:

- [PVBusBridge](#)

Ports for PVBusBridge

Table 4-24: Ports

Name	Protocol	Type	Description
control	PVBusBridgeControl	Slave	Control signal.
device	PVDevice	Master	Optimised connection out to devices.
dump_dmi	Signal	Slave	On the assert of this signal the bridge will dump dmi cache content into a csv file
pvbus_s	PVBus	Slave	Connection in from bus master.
reset	Signal	Slave	On the assert of this signal, a reset of the bus slave will be latched this is used by the bus deadlock detection logic.

4.2.23 SC_ClockSignal2ClockSignal

SystemC ClockSignal to ClockSignal converter. This model is written in LISA+.

SC_ClockSignal2ClockSignal contains the following CADI targets:

- [SC_ClockSignal2ClockSignal](#)

Ports for SC_ClockSignal2ClockSignal

Table 4-25: Ports

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

4.2.24 SGSignal2AMBAPVSignal

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



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

Ports for SGSignal2AMBAPVSignal

Table 4-26: 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.

4.2.25 SGStateSignal2AMBAPVSignalState

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



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

Ports for SGStateSignal2AMBAPVSignalState

Table 4-27: 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.

4.2.26 SGValue2AMBAPVValue

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



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

Ports for SGValue2AMBAPVValue

Table 4-28: 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.

4.2.27 SGValue2AMBAPVValue64

SystemGenerator Value_64 to AMBA-PV Value64 protocol converter. This model is written in LISA+.



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

Ports for SGValue2AMBAPVValue64

Table 4-29: 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.

4.2.28 SGValueState2AMBAPVValueState

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



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

Ports for SGValueState2AMBAPVValueState

Table 4-30: 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.

4.2.29 SGValueState2AMBAPVValueState64

SystemGenerator ValueState_64 to AMBA-PV ValueState64 protocol converter. This model is written in LISA+.



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

Ports for SGValueState2AMBAPVValueState64

Table 4-31: 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.

4.2.30 SMMUv3AEMIdentify2AMBAPVValue64

SMMUv3AEMIdentify to AMBA-PV Value64 protocol converter. This model is written in LISA+.

Ports for SMMUv3AEMIdentify2AMBAPVValue64

Table 4-32: Ports

Name	Protocol	Type	Description
identify	SMMUv3AEMIdentifyProtocol	Slave	SMMUv3AEMIdentifyProtocol input.
identify_reply	AMBAPVValue64	Slave	From SystemC.
identify_request	AMBAPVValue64	Master	To SystemC.

4.2.31 SystemC2Clock

Clock to SystemC Converter. This model is written in LISA+.

SystemC2Clock contains the following CADI targets:

- SystemC2Clock

Ports for SystemC2Clock

Table 4-33: 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.

Name	Protocol	Type	Description
rate_in_Hz_m	AMBAPVValueState64	Master	To SystemC.
set_clock_s	AMBAPVValue64	Slave	From SystemC.

4.2.32 SystemC2CprocBus

SystemCCoprocBusProtocol to CoprocBusProtocol converter. This model is written in LISA+.

SystemC2CprocBus contains the following CADI targets:

- SystemC2CprocBus

Ports for SystemC2CprocBus

Table 4-34: Ports

Name	Protocol	Type	Description
coproc_bus_m	CoprocBusProtocol	Master	-
sc_coproc_bus_s	SystemCCoprocBusProtocol	Slave	-

4.2.33 SystemC2CounterInterface

SystemC to CounterInterface Converter. This model is written in LISA+.

SystemC2CounterInterface contains the following CADI targets:

- SystemC2CounterInterface

Ports for SystemC2CounterInterface

Table 4-35: 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	-

4.2.34 SystemC2InstructionCount

SystemC to InstructionCount Converter. This model is written in LISA+.

SystemC2InstructionCount contains the following CADI targets:

- SystemC2InstructionCount



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

Ports for SystemC2InstructionCount

Table 4-36: 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.

4.2.35 SystemC2LCD

Converts SystemC to LCD protocol. This model is written in LISA+.

SystemC2LCD contains the following CADI targets:

- SystemC2LCD

Ports for SystemC2LCD

Table 4-37: 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.

4.2.36 SystemC2PChannel

SystemC to PChannel Converter. This model is written in LISA+.

SystemC2PChannel contains the following CADI targets:

- SystemC2PChannel

Ports for SystemC2PChannel

Table 4-38: Ports

Name	Protocol	Type	Description
pchannel	PChannel	Master	-
sc_pchannel	SystemCPChannel	Slave	-

4.2.37 SystemC2VirtualEthernet

SystemC to VirtualEthernet Converter. This model is written in LISA+.

SystemC2VirtualEthernet contains the following CADI targets:

- SystemC2VirtualEthernet

Ports for SystemC2VirtualEthernet

Table 4-39: Ports

Name	Protocol	Type	Description
virtualethernet_m	VirtualEthernet	Master	-
virtualethernet_s	SC_VirtualEthernet	Slave	-

4.2.38 SystemC2v7VGICConfig

Converts SystemC to v7_vgic_configuration_protocol. This model is written in LISA+.

SystemC2v7VGICConfig contains the following CADI targets:

- SystemC2v7VGICConfig

Ports for SystemC2v7VGICConfig

Table 4-40: 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.

Name	Protocol	Type	Description
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.

4.2.39 VECB2AMBAPVValue64

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

VECB2AMBAPVValue64 contains the following CADI targets:

- VECB2AMBAPVValue64

Ports for VECB2AMBAPVValue64

Table 4-41: 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.

4.2.40 VirtualEthernet2SystemC

VirtualEthernet to SystemC Converter. This model is written in LISA+.

VirtualEthernet2SystemC contains the following CADI targets:

- VirtualEthernet2SystemC

Ports for VirtualEthernet2SystemC

Table 4-42: Ports

Name	Protocol	Type	Description
virtualethernet_m	SC_VirtualEthernet	Master	-
virtualethernet_s	VirtualEthernet	Slave	-

4.2.41 v7VGICConfig2SystemC

Converts v7_vgic_configuration_protocol to SystemC. This model is written in LISA+.

v7VGICConfig2SystemC contains the following CADI targets:

- v7VGICConfig2SystemC

Ports for v7VGICConfig2SystemC

Table 4-43: 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.

4.3 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.

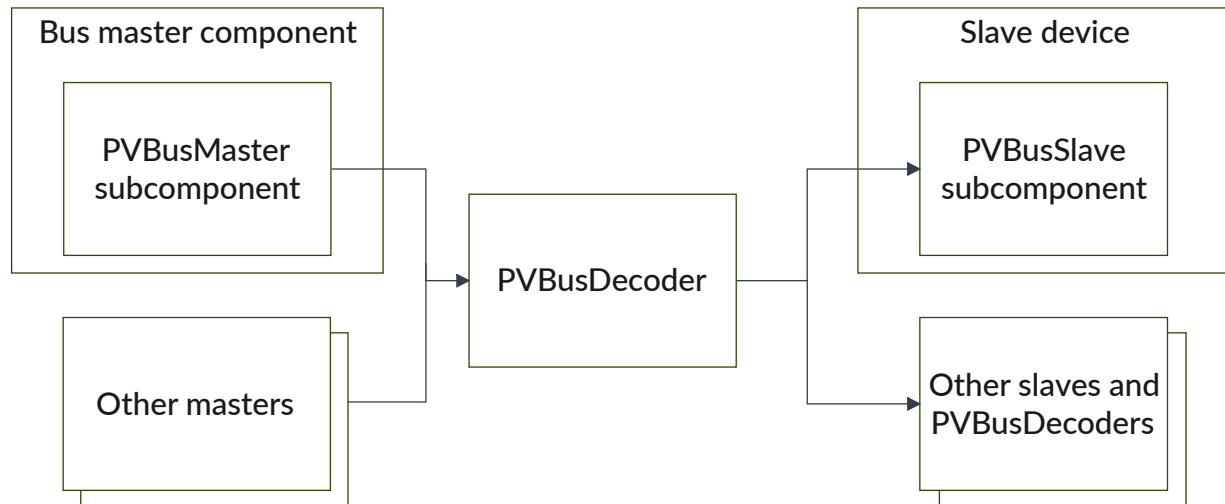
PVBus components provide functionally accurate communication between bus masters and slaves. They are not software implementations of specific hardware, but instead are abstract components that are required by the software model environment.

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 4-1: 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.

4.3.1 Labeller

This model is written in LISA+.

Labeller contains the following CADI targets:

- Labeller

Labeller contains the following MTI components:

- [PVBusMapper](#)

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 go between PVBUS Master and Slave ports.

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_HDLCD();
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 4-44: Ports

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

Parameters for Labeller

label

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

Type: int. Default value: 0x0.

4.3.2 LabellerForDMA330

This model is written in LISA+.

LabellerForDMA330 contains the following CADI targets:

- LabellerForDMA330

LabellerForDMA330 contains the following MTI components:

- [PVBUSMapper](#)

About LabellerForDMA330

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 go between PVBUS Master and Slave ports.

Ports for LabellerForDMA330

Table 4-45: Ports

Name	Protocol	Type	Description
pvbus_m	PVBUS	Master	Output with modified MasterID.

Name	Protocol	Type	Description
pvbuss_s	PVBus	Slave	Unmodified input.

Parameters for LabellerForDMA330

dma330_data_label

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.

Type: `int`. Default value: `0x0`.

dma330_discriminate_data_channels

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

Type: `bool`. Default value: `0x0`.

dma330_ns_instruction_label

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

Type: `int`. Default value: `0x0`.

dma330_s_instruction_label

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

Type: `int`. Default value: `0x0`.

4.3.3 LabellerForGPUProtMode

This model is written in LISA+.

LabellerForGPUProtMode contains the following CADI targets:

- LabellerForGPUProtMode

LabellerForGPUProtMode contains the following MTI components:

- [PVBusMapper](#)

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.

Ports for LabellerForGPUProtMode

Table 4-46: Ports

Name	Protocol	Type	Description
prot_mode	Signal	Slave	Input to determine whether output is supposed to be protected or not
pvbuss_m	PVBus	Master	Output with modified MasterID.
pvbuss_s	PVBus	Slave	Unmodified input.

Parameters for LabellerForGPUProtMode

gpu_id_normal

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

Type: `int`. Default value: `0x0`.

gpu_id_protected

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

Type: `int`. Default value: `0x0`.

4.3.4 MSIRewriter

Recognise writes to the GITS_TRANSLATER register and rewrite them to go to the GITS_TRANSLATE64R register. The DeviceID is expected to be in the bottom 32 bits of ExtendedID of the transaction. Debug transactions and reads are not rewritten. This component can also, optionally, apply a 16 bit 'label' to the top 16 bits of the MasterID in the same way that the Labeller component does. This model is written in LISA+.

MSIRewriter contains the following CADI targets:

- MSIRewriter

MSIRewriter contains the following MTI components:

- [MSIRewriter](#)
- [PVBusMapper](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

About MSIRewriter

MSIRewriter is a component that implements the functionality of the MSI-64 Encapsulator in GIC IP, for example GIC-700.

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.

MasterID, ExtendedID, and UserFlags

These tables show how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

Table 4-47: pvbus_s port

PVBus attribute	Bits used	Property encoded	Notes
MasterID	-	-	Not used.
ExtendedID	Bits[63:32]	Stream ID	-
	Bits[31:0]	Device ID	-

PVBus attribute	Bits used	Property encoded	Notes
UserFlags	-	-	Not used.

Table 4-48: pvbush_m port

PVBus attribute	Bits used	Property encoded	Notes
MasterID	Bits[31:16]	Optional 16-bit label.	For usage, see the <code>label</code> parameter.
ExtendedID	-	-	Not used.
UserFlags	-	-	Not used.

Ports for MSIRewriter

Table 4-49: Ports

Name	Protocol	Type	Description
pvbush_m	PVBus	Master	-
pvbush_s	PVBus	Slave	-

Parameters for MSIRewriter

enable_rewriting

Enable rewriting.

Type: `bool`. Default value: `0x1`.

GITS_TRANSLATE64R_OFFSET

It is an offset from ITS0-Base.

Type: `int`. Default value: `0x10048`.

ITS0-base

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

Type: `int`. Default value: `0x0`.

label

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).

Type: `int`. Default value: `0xffffffff`.

log

Log level, 0 is off.

Type: `int`. Default value: `0x0`.

Related information

[GIC architecture specification version 3 and version 4](#)

[GIC-700 Technical Reference Manual](#)

4.3.5 PASSwitch

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

Ports for PASSwitch

Table 4-50: Ports

Name	Protocol	Type	Description
control	PASSwitchControl	Slave	Controls routing of transactions
pdbus_m[4]	PVBus	Master	Manager ports of PASSwitch
pdbus_s	PVBus	Slave	Subordinate port of PASSwitch

4.3.6 PVBus4KBTo1KBSplitter

Takes 4KB of address range input on slave port and routes each 1KB to four different master ports. This model is written in LISA+.

PVBus4KBTo1KBSplitter contains the following CADI targets:

- PVBus4KBTo1KBSplitter

PVBus4KBTo1KBSplitter contains the following MTI components:

- [PVBusMaster](#)
- [PVBusSlave](#)

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
- 0xc00-0xffff

This overcomes a limitation of PVBus which only allows components to be attached to memory addresses that are a multiple of 4KB in size.

Ports for PVBus4KBTo1KBSplitter

Table 4-51: Ports

Name	Protocol	Type	Description
pdbus_m[4]	PVBus	Master	The four downstream ports to be connected to peripherals. Each port covers 1KiB of the address space. Output address on each port will be in the range 0x0 - 0x03FF.

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

4.3.7 PVBusCache

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

Ports for PVBusCache

Table 4-52: 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.
device	PVBusCacheDevice	Master	Connection out to cache device.

4.3.8 PVBusDecoder

A PVBusDecoder allows bus transactions to be routed to one of many slaves, based on the address given in the transaction. This model is written in C++.

About PVBusDecoder

Each slave connection is associated with a specific address range on the `pvbus_m_range` port. In LISA+, the syntax for this is:

```
decoder.pvbus_m_range[start..end] = slave.pvbus
```

The values for `start` (inclusive) and `end` (inclusive) must specify a 4KB-aligned region of a multiple of 4K bytes. You can specify an address range for the slave, where the decoder remaps addresses into the appropriate range. The default address range for a slave is $[0-(\text{sizeofMasterRange} - 1)]$.

Ports for PVBusDecoder

Table 4-53: Ports

Name	Protocol	Type	Description
pvbus_m_range	PVBus	Master	Specifies the address range for the bus master. The range must be 4KB aligned and a multiple of 4KB in size. If the address range is larger than the size of the slave device, the slave is aliased.
pvbus_s	PVBus	Slave	Accepts incoming transactions. Connect this port to a bus master, or to the output of another bus decoder.

4.3.9 PVBusExclusiveMonitor

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

PVBusExclusiveMonitor contains the following CADI targets:

- PVBusExclusiveMonitor

PVBusExclusiveMonitor contains the following MTI components:

- [PVBusExclusiveMonitor](#)
- [PVBusMapper](#)

Ports for PVBusExclusiveMonitor

Table 4-54: Ports

Name	Protocol	Type	Description
excl_cleared	Signal	Master	Exclusive monitor clear signal port.
pvbus_m	PVBus	Master	Bus master port.
pvbus_s	PVBus	Slave	Bus slave port.

Parameters for PVBusExclusiveMonitor

exclusive_monitor0.apply_access_width_criteria_to_non_excl_stores

Apply the given exclusive store width matching criteria to non-exclusive stores.
Type: `bool`. Default value: `0x1`.

exclusive_monitor0.clear_on_strex_address_mismatch

Whether monitor is cleared when strex fails due to address mismatch.
Type: `bool`. Default value: `0x1`.

exclusive_monitor0.enable_component

Enable component.
Type: `bool`. Default value: `0x1`.

exclusive_monitor0.exclusive_monitor_clear_on_atomic_from_same_master

Monitor atomics from the same master.
Type: `bool`. Default value: `0x1`.

exclusive_monitor0.log2_granule_size

log2 of address granule size.
Type: `int`. Default value: `0x0`.

exclusive_monitor0.match_access_width

Fail STREX if not the same access width as LDREX.
Type: `bool`. Default value: `0x0`.

exclusive_monitor0.match_secure_state

Treat the secure state like an address bit.
Type: `bool`. Default value: `0x1`.

exclusive_monitor0.monitor_access_level

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

Type: `int`. Default value: `0x0`.

exclusive_monitor0.monitor_non_excl_stores

Monitor non-exclusive stores from the same master.

Type: `bool`. Default value: `0x0`.

exclusive_monitor0.number_of_monitors

Number of monitors.

Type: `int`. Default value: `0x8`.

exclusive_monitor0.shareability_domain

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

Type: `int`. Default value: `0x3`.

4.3.10 PVBusExclusiveSquasher

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

PVBusExclusiveSquasher contains the following CADI targets:

- PVBusExclusiveSquasher

PVBusExclusiveSquasher contains the following MTI components:

- [PVBusMapper](#)

Ports for PVBusExclusiveSquasher

Table 4-55: Ports

Name	Protocol	Type	Description
<code>pvbus_m</code>	PVBus	Master	Bus master port.
<code>pvbus_s</code>	PVBus	Slave	Bus slave port.

4.3.11 PVBusLogger

A PVBusLogger has a slave and a master port and traffic is passed straight through. All traffic is logged using an MTI trace event. This model is written in C++.

PVBusLogger contains the following CADI targets:

- PVBusLogger

PVBusLogger contains the following MTI components:

- [PVBusLogger](#)

- [PVBusMapper](#)

Ports for PVBusLogger

Table 4-56: Ports

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Bus master port.
pvbus_s	PVBus	Slave	Bus slave port.

Parameters for PVBusLogger

pvbuslogger.trace_debug

Enable tracing of debug transactions.
Type: bool. Default value: 0x0.

pvbuslogger.trace_snoops

Enable tracing of ACE snoop requests.
Type: bool. Default value: 0x0.

4.3.12 PVBusMapper

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

PVBusMapper contains the following MTI components:

- [PVBusMapper](#)

About PVBusMapper

This component performs like `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 4-57: 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.

4.3.13 PVBusMaster

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

PVBusMaster contains the following MTI components:

- [PVBusMaster](#)

About PVBusMaster

If you want a component to act as a bus master, instantiate a `PVBusMaster` subcomponent and then use its control port to create one or more transaction generators to generate transactions on the bus. `$PVLIB_HOME/examples/LISA/BusComponents/DmaTransfer.lisa` is an example component that shows this.

Ports for PVBusMaster

Table 4-58: Ports

Name	Protocol	Type	Description
control	PVTransactionMaster	Slave	Enables the owning component to instantiate <code>pv::TransactionGenerator</code> objects.
pvbus_m	PVBus	Master	Sends out generated transactions to the bus.
reset	Signal	Slave	On the de-assert of this signal, a reset of the bus master will be latched this is used by the bus deadlock detection logic.

Related information

[TransactionGenerator efficiency considerations](#) on page 63

4.3.14 PVBusModifier

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

PVBusModifier contains the following MTI components:

- [PVBusMapper](#)



PVBusModifierx2 is an identical component, except it has two downstream ports.

Ports for PVBusModifier

Table 4-59: Ports

Name	Protocol	Type	Description
control	PVBusMapperControl	Master	Configuration port to determine mappings.
pvbus_m	PVBus	Master	Bus master port.

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	Bus slave port.
reset	Signal	Slave	Reset signal.

4.3.15 PVBusRouter

Allow transactions to be routed arbitrarily. This model is written in LISA+.

PVBusRouter contains the following CADI targets:

- PVBusRouter

PVBusRouter contains the following MTI components:

- [PVBusMapper](#)

Ports for PVBusRouter

Table 4-60: Ports

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

4.3.16 PVBusSlave

A PVBusSlave handles incoming transactions, and handles support for mapping regions of device address space to work as RAM/ROM/device memory. This model is written in C++.

PVBusSlave contains the following MTI components:

- [PVBusSlave](#)

Changes in 11.24.4

Parameters added:

- read_latency
- write_latency

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.

By default, the PVBusSlave translates all transactions into `read()` and `write()` requests on the device port.

The control port enables you to configure the PVBusSlave for a device. This lets you define the behavior of the different regions of the address space on the device. You can configure regions separately for read and write, at a 4KB granularity. This permits you to set memory-like, device-like, abort, or ignore access address regions.

Transactions to memory-like regions are handled internally by the PVBusSlave subcomponent. Abort and ignore regions are also handled by the PVBusSlave.

Transactions to device-like regions are forwarded to the device port. Your device must implement the `read()` and `write()` methods on the device port if any regions are configured as device-like.

This component typically does not significantly affect the performance of a PV system. However, correct implementation of the PVBusSlave component is critical to the performance of the overall PV system. For example, routing a request to a PVDevice port is slower than letting the PVBusSlave component handle the request internally. Arm recommends using the internal support for memory-like regions where possible.

Ports for PVBusSlave

Table 4-61: 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.

4.3.17 PVMemoryProtectionEngine

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

PVMemoryProtectionEngine contains the following CADI targets:

- [PVMemoryProtectionEngine](#)

PVMemoryProtectionEngine contains the following MTI components:

- [PVBusMapper](#)
- [PVMemoryProtectionEngine](#)

About PVMemoryProtectionEngine

PVMemoryProtectionEngine is a simplified implementation of a Memory Protection Engine (MPE) component as described in *Realm Management Extension (RME) System Architecture*.

PVMemoryProtectionEngine features:

- Supports 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 4-62: 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

Encryption block size, supported sizes are 1 or 4096.

Type: `int`. Default value: `0x1000`.

corruption_strategy

Corruption strategy (0 - fill with constants per-old-encryption-context, 1 - fill with constants per-new-encryption-context, 2 - random data).

Type: `int`. Default value: `0x0`.

enable

Enables the Memory Protection Engine.

Type: `bool`. Default value: `0x0`.

ignore_mecid

Ignore MECID during encryption key calculation.

Type: `bool`. Default value: `0x0`.

non_secure_pas_enc_key

Non-secure PAS encryption key.

Type: `int`. Default value: `0x22`.

output_attributes_parameter_of_core

Encoding of various attributes on the bus.

Type: `string`. Default value: "ExtendedID[62:55]=MPAM_PMG,
ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS".

realm_pas_enc_key

Realm PAS encryption key.

Type: `int`. Default value: `0x88`.

root_pas_enc_key

Root PAS encryption key.

Type: `int`. Default value: `0x44`.

secure_pas_enc_key

Secure PAS encryption key.

Type: `int`. Default value: `0x11`.

Related information

[Arm Realm Management Extension \(RME\) System Architecture](#)

4.3.18 PVWriteBuffer

The PVWriteBuffer subcomponent buffers PVBus transactions. This model is written in C++.

PVWriteBuffer contains the following MTI components:

- [PVBusMapper](#)
- [PVWriteBuffer](#)

Ports for PVWriteBuffer

Table 4-63: Ports

Name	Protocol	Type	Description
barrier_notify_s	PVWriteBuffer_BarrierPort	Slave	Barrier notification input.
clk_in	ClockSignal	Slave	Clock input.
pdbus_m	PVBus	Master	Master connection to memory bus.
pdbus_s	PVBus	Slave	Slave connection for transactions to be buffered.
reset_in	Signal	Slave	Reset input.
serror_notify_m	PVWriteBuffer_SErrorPort	Master	SError output generation.

4.3.19 SimplePVBusMaster

Component to generate PVTransactions with configurable attributes and address. This model is written in LISA+.

SimplePVBusMaster contains the following CADI targets:

- ClockTimerThread64
- SchedulerThread
- SchedulerThreadEvent
- SimplePVBusMaster

SimplePVBusMaster contains the following MTI components:

- [PVBusMaster](#)
- [PVBusSlave](#)

Ports for SimplePVBusMaster

Table 4-64: Ports

Name	Protocol	Type	Description
pdbus_m	PVBus	Master	Output of generated transactions.
pdbus_s	PVBus	Slave	-

Parameters for SimplePVBusMaster

verbose

verbose.

Type: bool. Default value: 0x0.

4.3.20 TZSwitch

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

TZSwitch contains the following CADI targets:

- TZSwitch

TZSwitch contains the following MTI components:

- [PVBusMapper](#)

About TZSwitch

The default behavior is to forward secure transactions to `pvtbus_port_a`, and normal transactions to `pvtbus_port_b`.

You must only use TZSwitches if the routing decisions change infrequently, for example as part of a memory remap.

The `secure` and `normal` parameters control the initial state of this component:

- | | |
|----------|--|
| 0 | Ignore these transactions. |
| 1 | Forward the transactions to <code>pvtbus_port_a</code> . |
| 2 | Forward the transactions to <code>pvtbus_port_b</code> . |
| 3 | Generate an abort for these transactions. |

The numbers used for initial configuration are not the same as the enumeration constants used to control routing at runtime.

Ports for TZSwitch

Table 4-65: Ports

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

Parameters for TZSwitch

`tzswitch_0.normal`

Normal Port.

Type: `int`. Default value: `0x2`.

`tzswitch_0.secure`

Secure Port.

Type: `int`. Default value: `0x1`.

4.4 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.

4.4.1 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++.

ClockDivider contains the following CADI targets:

- ClockDivider

ClockDivider contains the following MTI components:

- [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.

For an example of the use of ClockDividers, see the `vEMotherBoard.lisa` component in the `$PVLIB_HOME/examples/LISA/FVP_VE/LISA/` directory of your Fast Models distribution.

ClockDivider parameters

The ClockDivider parameters are not exposed by CADI. You can set them from LISA but not from SystemC:

mul

Clock rate multiplier.

Type: `uint64_t`. Default value: 1.

div

Clock rate divider.

Type: `uint64_t`. Default value: 1.

Ports for ClockDivider

Table 4-66: Ports

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

4.4.2 ClockTimer

A ClockTimer provides support for counting a number of ticks at the rate determined by the input clock. When the countdown reaches zero, the timer can send a signal to another component. That component can return a value that will cause 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. To use a ClockTimer connect the ClockTimer's 'timer_callback' port to a slave port that implements the 'signal()' behaviour and connect a clock signal to the clk_in port. Use the 'timer_control' port to start the timer counting down for a given number of ticks. This model is written in C++.

About ClockTimer

This component provides a mechanism for other components to schedule a callback after a number of ticks at a given ClockSignal rate.

An active ClockTimer component incurs no simulation overhead. For best performance, avoid having your performance-critical code frequently cancel timers or query the number of remaining ticks.

Ports for ClockTimer

Table 4-67: Ports

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

4.4.3 ClockTimer64

A ClockTimer64 provides support for counting a number of ticks at the rate determined by the input clock. When the countdown reaches zero, the timer can send a signal to another component. That component can return a value that will cause 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. This model is written in C++.

About ClockTimer64

This component provides a mechanism for other components to schedule a callback after a number of ticks at a given ClockSignal rate.

An active ClockTimer64 component incurs no simulation overhead. For best performance, avoid having your performance-critical code frequently cancel timers or query the number of remaining ticks.

Ports for ClockTimer64

Table 4-68: 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.

4.4.4 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 mean 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 which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64). This model is written in LISA+.

ClockTimerThread contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- SchedulerThread
- SchedulerThreadEvent

Ports for ClockTimerThread

Table 4-69: Ports

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

4.4.5 ClockTimerThread64

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. 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 which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64). This model is written in LISA+.

ClockTimerThread64 contains the following CADI targets:

- ClockTimerThread64
- SchedulerThread
- SchedulerThreadEvent

Ports for ClockTimerThread64

Table 4-70: 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.

4.4.6 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 1Hz.

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

Ports for MasterClock

Table 4-71: Ports

Name	Protocol	Type	Description
clk_out	ClockSignal	Master	Master clock rate.

4.5 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 sacrifice timing accuracy. They also use efficient PV bus models to enable fast access to memory and devices.

CT components implement most of the processor features but differ in some ways:

- 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* (PV) 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.

4.5.1 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 4-72: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

AEMv8RMPCT contains the following CADI targets:

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

AEMv8RMPCT contains the following MTI components:

- [ARM_AEMv8-R_MP](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

About AEMv8RMPCT

AEMv8RMPCT allows you to target AArch32 or AArch64, RAS, VFP, EL2, and other Arm®v8-R features.

Table 4-73: 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
Arm®v8.5 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 4-74: 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 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.
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

Name	Protocol	Type	Description
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	-
cti1extin[4]	Signal	Slave	-
cti1extout[4]	Signal	Master	-
cti2extin[4]	Signal	Slave	-
cti2extout[4]	Signal	Master	-
cti3extin[4]	Signal	Slave	-
cti3extout[4]	Signal	Master	-
ctidbgirq[4]	Signal	Master	-
dbgen[4]	Signal	Slave	-
dbgnopwrdown[4]	Signal	Master	These signals relate to core power down.
dbgpwrdownack[4]	Signal	Master	Debug power down acknowledge.
dbgpwrdownreq[4]	Signal	Slave	Debug power down request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
ext_slave_s	PVBus	Slave	External Slave port. Equivalent to AXIS port.
external_trace_reset[4]	Signal	Slave	ETMv4 External Trace Reset signal.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
flash_m[4]	PVBus	Master	Flash Port
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
hiden[4]	Signal	Slave	External debug interface.
hniden[4]	Signal	Slave	External debug interface.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.

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 and l2cache
llpp_m[4]	PVBus	Master	LLPP (Low-Latency Peripheral Port).
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg[4]	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	-
romaddr	Value_64	Slave	Debug ROM base address.
romaddrv	Signal	Slave	Debug ROM base address valid.
rvbar[4]	Value_64	Slave	Reset vector base address.
sei[4]	Signal	Slave	Per core System Error physical pins
smpnamp[4]	Signal	Master	This signals AMP or SMP mode for each core
spiden[4]	Signal	Slave	Secure invasive debug enable.
spniden[4]	Signal	Slave	Secure non-invasive debug enable.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state
teinit[4]	Signal	Slave	This signal provides default exception handling state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trace_unit_reset[4]	Signal	Slave	ETMv4 Trace Unit Reset signal.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARMAEMv8-R_MP

cpu0.aarch32_reset_from_impdef_addr

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

Type: `bool`. Default value: `0x1`.

cpu0.ase-present

Set whether the model has been built with NEON support.

Type: `bool`. Default value: `0x1`.

cpu0.CFGEND

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

Type: `bool`. Default value: `0x0`.

cpu0.clock_divider

Clock divider ratio for asymmetric MP clocking.

Type: `int`. Default value: `0x1`.

cpu0.clock_multiplier

Clock divider ratio for asymmetric MP clocking.

Type: `int`. Default value: `0x1`.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: `bool`. Default value: `0x0`.

cpu0.CP15SDISABLE2

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

Type: `bool`. Default value: `0x0`.

cpu0.crypto_aes

AES instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 2, AES and PMULL instructions implemented (FEAT_AES, FEAT_PMULL).

Type: `int`. Default value: `0x2`.

cpu0.crypto_sha1

SHA-1 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA1 instructions implemented (FEAT_SHA1).

Type: `int`. Default value: `0x1`.

cpu0.crypto_sha256

SHA-256 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA256 instructions implemented (FEAT_SHA256).

Type: `int`. Default value: `0x1`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.cti-intack_mask

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

Type: `int`. Default value: `0x1`.

cpu0.cti-number_of_claim_bits

Number of implemented bits in CTICLAIMSET.

Type: `int`. Default value: `0x0`.

cpu0.cti-number_of_triggers

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

Type: `int`. Default value: `0x8`.

cpu0.dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

cpu0.DCZID-log2-block-size

Log2 of the block size cleared by DC ZVA instruction (as read from DCZID_EL0).

Type: `int`. Default value: `0x8`.

cpu0.DCZVA_single_write

Execute the DCZVA as a single write.

Type: `bool`. Default value: `0x0`.

cpu0.enable_crc32

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

Type: `int`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.etm-present

Set whether the model has ETM support.

Type: `bool`. Default value: `0x1`.

cpu0.flash.enable

Enable flash by default after reset.

Type: `bool`. Default value: `0x0`.

cpu0.force-fpsid

Override the FPSID value.

Type: `bool`. Default value: `0x0`.

cpu0.force-fpsid-value

Value to override the FPSID value to.

Type: `int`. Default value: `0x0`.

cpu0.has_hcptr_tase

If false, HCPTR.TASE is RES0.

Type: `bool`. Default value: `0x1`.

cpu0.icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

cpu0.llpp.base

Sets the base address of Low Latency Peripheral Port.

Type: `int`. Default value: `0x0`.

cpu0.llpp.size

Sets the size of LLPP(in bytes).

Type: `int`. Default value: `0x1000`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.MPIDR-override

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

Type: `int`. Default value: `0x0`.

cpu0.number-of-breakpoints

Number of breakpoints.

Type: `int`. Default value: `0x10`.

cpu0.number-of-context-breakpoints

Number of breakpoints that are context aware.

Type: `int`. Default value: `0x10`.

cpu0.number-of-watchpoints

Number of watchpoints.

Type: `int`. Default value: `0x10`.

cpu0.operation_bandwidth

Operation width for ARMv8.4 PMU extension.

Type: `int`. Default value: `0x1`.

cpu0.RVBAR

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

cpu0.RVBAR32

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

Type: `int`. Default value: `0x0`.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.
Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.
Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.
Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xffffffff.

cpu0.semihosting-prefix

Prefix semihosting output with target instance name.
Type: bool. Default value: 0x0.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: int. Default value: 0xffffffff.

cpu0.semihosting-stderr_istty

Result for semihost istty call when argument is stderr.
Type: bool. Default value: 0x1.

cpu0.semihosting-stdin_istty

Result for semihost istty call when argument is stdin.
Type: bool. Default value: 0x1.

cpu0.semihosting-stdout_istty

Result for semihost istty call when argument is stdout.
Type: bool. Default value: 0x1.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: int. Default value: 0x3c.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.semihosting-use_stderr

Send stderr from the simulated process to host stderr.

Type: `bool`. Default value: `0x0`.

cpu0.SMPnAMP

Enable broadcast messages necessary for correct SMP operation at reset.

Type: `bool`. Default value: `0x1`.

cpu0.tcm-present

Disables the TCMs.

Type: `bool`. Default value: `0x0`.

cpu0.tcm-supports-exclusive

Whether TCM supports exclusive access.

Type: `bool`. Default value: `0x0`.

cpu0.tcm.a.base

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

Type: `int`. Default value: `0x0`.

cpu0.tcm.a.enable

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

Type: `bool`. Default value: `0x0`.

cpu0.tcm.a.size

Sets the size of the ATCM(in bytes). The size should be aligned with 4KB. The lower 12 bits will be masked out.

Type: `int`. Default value: `0x4000`.

cpu0.tcm.a.stretch_clk

Whether ATCM clock stretched to occupy full cycle.

Type: `bool`. Default value: `0x0`.

cpu0.tcm.a.wait

ATCM accesses wait states.

Type: `int`. Default value: `0x0`.

cpu0.tcm.b.base

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

Type: `int`. Default value: `0x0`.

cpu0.tcm.b.enable

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

Type: `bool`. Default value: `0x0`.

cpu0.tcm.b.size

Sets the size of the BTCM(in bytes). The size should be aligned with 4KB. The lower 12 bits will be masked out.

Type: `int`. Default value: `0x4000`.

cpu0.tcm.b.stretch_clk

Whether BTCM clock stretched to occupy full cycle.

Type: `bool`. Default value: `0x0`.

cpu0.tcm.b.wait

BTCM accesses wait states.

Type: `int`. Default value: `0x0`.

cpu0.tcm.c.base

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

Type: `int`. Default value: `0x0`.

cpu0.tcm.c.enable

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

Type: `bool`. Default value: `0x0`.

cpu0.tcm.c.size

Sets the size of the CTCM(in bytes). The size should be aligned with 4KB. The lower 12 bits will be masked out.

Type: `int`. Default value: `0x2000`.

cpu0.tcm.c.stretch_clk

Whether CTCM clock stretched to occupy full cycle.

Type: `bool`. Default value: `0x0`.

cpu0.tcm.c.wait

CTCM accesses wait states.

Type: `int`. Default value: `0x0`.

cpu0.TEINIT

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

Type: `bool`. Default value: `0x0`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0x1000`.

cpu0.unpredictable_WPMASKANDBAS

Constrained unpredictable handling of watchpoints when mask and BAS fields specified. 0, IGNOREMASK. 1, IGNOREBAS (default). 2, REPEATBAS8. 3, REPEATBAS.

Type: `int`. Default value: `0x1`.

cpu0.vfp-dp-present

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

Type: `bool`. Default value: `0x1`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool. Default value: 0x0.

cpu0.vfp-present

Set whether the model has VFP support.

Type: bool. Default value: 0x1.

cpu0.vfp-traps

Implement support for trapping floating-point exceptions.

Type: bool. Default value: 0x1.

cpu0.vfp-traps-show-all

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

Type: bool. Default value: 0x0.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: bool. Default value: 0x0.

cpu0.wfet_early_or_delayed_timeout

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

Type: int. Default value: 0x0.

cpu0.wfit_early_or_delayed_timeout

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

Type: int. Default value: 0x0.

Parameters for Cluster_ARMAEMv8-R_MP**abort_execution_from_device_memory**

Execution from device memory generates a prefetch abort.

Type: bool. Default value: 0x0.

ADFSR-AIFSR-implemented

ADFSR and AIFSR are implemented.

Type: bool. Default value: 0x0.

advsimd_overread

AdvSIMD element load operations access all bytes of a 16-byte aligned window, even in Device memory.

Type: bool. Default value: 0x0.

AIDR

Value of AIDR_EL1 register.

Type: int. Default value: 0x0.

align_pc_on_branch_to_unaligned_pc_aarch32

Force PC align for branches to an unaligned PC counter in A32 state.

Type: bool. Default value: 0x0.

align_pc_on_debug_exit_to_aarch32

Exit to AARCH32 state from debug state forces pc bit0 to 0.

Type: bool. Default value: 0x0.

align_pc_on_illegal_exception_return_to_aarch32

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

Type: bool. Default value: 0x1.

AMIIDR

Value of AMU Implementation Identification Register.

Type: int. Default value: 0x43b.

AMPIDR

Value of AMU Peripheral Identification Register.

Type: int. Default value: 0x4000bb000.

amu_aux_type_fixed

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.

Type: string. Default value: "".

amu_num_auxiliary_counters

Number of AMU auxiliary counters implemented.

Type: int. Default value: 0x0.

apshr_read_restrict

At EL0, unknown bits of APSR are RAZ.

Type: bool. Default value: 0x0.

auxilliary_feature_register0

Value of AFR0 ID register.

Type: int. Default value: 0x0.

BPIMVA_causes_translation_lookup

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

Type: bool. Default value: 0x0.

branch-predictor-clear-policy

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.

Type: int. Default value: 0x2.

branch-predictor-supported-ops

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.

Type: int. Default value: 0x1.

BROADCASTATOMIC

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

Type: bool. Default value: 0x1.

BROADCASTATOMICL

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

Type: bool. Default value: 0x1.

bus_protection_enable_at_reset

Enable TCM, L1Cache bus protection after reset.

Type: bool. Default value: 0x0.

cache-log2linelen

Log2 of the cache line length in bytes.

Type: int. Default value: 0x6.

cache_maintenance_hits_watchpoints

DCIMVA operations executed in AArch32 modes hit watchpoints.

Type: bool. Default value: 0x0.

CCSIDR-L1D_override

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

Type: int. Default value: 0x0.

CCSIDR-L1I_override

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

Type: int. Default value: 0x0.

CCSIDR-L2_override

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

Type: int. Default value: 0x0.

CCSIDR-L3_override

If nonzero, allow L3 selection in CSSELR and present this value in CCSIDR (this is cosmetic and does not affect cache behaviour).

Type: int. Default value: 0x0.

CFGTFPEN_pin_reset

CFGTFPEN Configuration pin at reset for bitfield IMP_MEMPROTCTLR_EL1.TFPEN.

Type: bool. Default value: 0x0.

changing_block_size_without_bbm_support

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

Type: int. Default value: 0x0.

check_memory_attributes

Detect and report TLB use of conflicting memory attributes for views of the same physical address.

Type: bool. Default value: 0x1.

clear_reg_top_eret

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.

Type: `int`. Default value: `0x1`.

clear_reg_top_set

Whether to clear upper 32-bits of the Xn register when corresponding AArch32 register is set via CADI/Iris.

Type: `bool`. Default value: `0x1`.

cluster_utid

Unique cluster transaction identifier for interconnect protection.

Type: `int`. Default value: `0x0`.

core_cache_protection

`core_cache_protection` can change `ERROFR`, `ERROPFGF` and `ERROPFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int`. Default value: `-0x1`.

cpacr_trcds_behaviour

Behaviour of CPACR.TRCDIS/NSACR.NSTRCDIS when there is no CP14 ETM interface. 0, RAZ/WI. 2, implemented.

Type: `int`. Default value: `0x2`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

CPUCFR

Value of CPU Configuration Register.

Type: `int`. Default value: `0x0`.

CTIPIDR

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

Type: `int`. Default value: `0x0`.

DBGBCR_BT_applies_RES0_before_valid_check

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

Type: `bool`. Default value: `0x1`.

dbgitr_buffer_size

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

Type: `int`. Default value: `0x0`.

DBGPIDR

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

Type: `int`. Default value: `0x0`.

dbgxvr_ress_is_stateful

Whether DBGWVR/DBGBVR.RESS returns last written value. if set to false, RESS returns sign extended value.

Type: `bool`. Default value: `0x0`.

dc_fault_unaligned_s1_device_s2_fwb

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.

Type: `bool`. Default value: `0x0`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_bus_width_in_bytes

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

Type: `int`. Default value: `0x8`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

`dcache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-ways`

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

Type: `int`. Default value: `0x2`.

`dcache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-write_bus_width_in_bytes`

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

Type: `int`. Default value: `0x8`.

`dcache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcimva_requires_s2_write_permissions`

Data-cache invalidate by MVA operations require stage 2 write permission (virtualised AArch32 guest).

Type: `bool`. Default value: `0x0`.

`debug_rom_is_class_9`

If true, present a debug ROM table as a class 9 device. Otherwise, use a class 1 ROM table.

Type: `bool`. Default value: `0x0`.

`debug_rom_is_flat`

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

Type: `bool`. Default value: `0x0`.

`def_mem_map`

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.

Type: `string`. Default value: `"{ "*" : "NORMAL INC ONC OSH" }`.

def_mem_map_file_path

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.

Type: `string`. Default value: `""`.

default_inner_shareable

shareability for default memory map regions which are shareable.

Type: `bool`. Default value: `0x0`.

delay_serror

Add a propagation delay of serror signal into the core.

Type: `int`. Default value: `0x0`.

dic-spi_count

Number of shared peripheral interrupts implemented.

Type: `int`. Default value: `0x40`.

disable_impdef_abort_on_ic_maintenance

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

Type: `bool`. Default value: `0x0`.

disable_sve_plugin

If true, SVE will not be implemented in this processor even if the plugin is loaded (FEAT_SVE).

Type: `bool`. Default value: `0x0`.

disable_unknown_update_event_on_reset

Disables SYSREG_UPDATE event notification on reset for the registers whose bitfields are all reserved or resets to architecturally unknown value.

Type: `bool`. Default value: `0x0`.

dsb_accumulate_threshold

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

Type: `int`. Default value: `0x100`.

el0_can_access_imp_def_functionality

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

Type: `bool`. Default value: `0x0`.

el3_trap_priority_when_secure_debug_disabled

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

Type: `bool`. Default value: `0x0`.

enable_address_contig_check

Check the input address range for the table entries that have the contiguous hint bit set.

Type: `bool`. Default value: `0x0`.

enable_lock_step

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

Type: `bool`. Default value: `0x0`.

enable_tlb_contig_check

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

Type: `bool`. Default value: `0x0`.

enhanced_pac2_level

Implements Enhanced PAC2. 0: No EnhancedPAC2, 1: EnhancedPAC2 Only, 2: EnhancedPAC2 with FPAC, 3: EnhancedPAC2 with FPACCombined.

Type: `int`. Default value: `0x0`.

error_record_feature_register

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,"INJ":0x0,"FRX":0x0,"UC":0x0,"UEU":0x0,"UER":0x0,"UEO":0x0,"DE":0x0,"CI":0x0,"TS":0x0,"Visibility":0x0,"Core":0x0,"Cluster":0x0}]
```

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".

Type: `string`. Default value: `""`.

error_record_feature_register_json_file

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

Type: `string`. Default value: `""`.

ERXMISC0_mask

Write Mask for ERXMISC0 RAS Register.

Type: `int`. Default value: `0x0`.

exception_catch_before_software_step

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.

Type: `bool`. Default value: `0x1`.

exception_catch_type

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.

Type: `int`. Default value: `0x0`.

exclusive_monitor_clear_on_atomic_from_same_master

Exclusive monitors in the cluster will be cleared by a atomic by the same master to the monitored address.

Type: `bool`. Default value: `0x1`.

exclusive_monitor_clear_on_store_from_same_master

Exclusive monitors in the cluster will be cleared by a store by the same master to the monitored address.

Type: `bool`. Default value: `0x1`.

exclusive_monitor_clear_on_strex_address_mismatch

Exclusive monitors in the cluster will be cleared when a strex fails because the address does not match.

Type: `bool`. Default value: `0x1`.

exclusive_monitor_clear_on_strex_success

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

Type: `bool`. Default value: `0x1`.

exercise_stxr_fail

Reject a pseudo-random majority of exclusive store instructions.

Type: `bool`. Default value: `0x0`.

ext_abort_device_GRE_prefetch_ras_index

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`).

Type: `int`. Default value: `-0x1`.

ext_abort_device_GRE_prefetch_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_device_GRE_read_is_critical

Critical reporting of device-GRE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_device_GRE_read_is_sync

Synchronous reporting of device-GRE read external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_device_read_is_sync`.

Type: `int`. Default value: `0x2`.

ext_abort_device_GRE_read_ras_index

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`).

Type: `int`. Default value: `-0x1`.

ext_abort_device_GRE_read_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_device_GRE_write_is_critical

Critical reporting of device-GRE write external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_device_GRE_write_is_sync

Synchronous reporting of device-GRE write external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_device_write_is_sync`.

Type: `int`. Default value: `0x2`.

ext_abort_device_GRE_write_ras_index

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).

Type: int. Default value: -0x1.

ext_abort_device_GRE_write_ras_type

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.

Type: int. Default value: -0x1.

ext_abort_device_nGRE_prefetch_ras_index

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).

Type: int. Default value: -0x1.

ext_abort_device_nGRE_prefetch_ras_type

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.

Type: int. Default value: -0x1.

ext_abort_device_nGRE_read_is_critical

Critical reporting of device-nGRE read external aborts.

Type: bool. Default value: 0x0.

ext_abort_device_nGRE_read_is_sync

Synchronous reporting of device-nGRE read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext_abort_device_read_is_sync.

Type: int. Default value: 0x2.

ext_abort_device_nGRE_read_ras_index

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).

Type: int. Default value: -0x1.

ext_abort_device_nGRE_read_ras_type

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.

Type: int. Default value: -0x1.

ext_abort_device_nGRE_write_is_critical

Critical reporting of device-nGRE write external aborts.

Type: bool. Default value: 0x0.

ext_abort_device_nGRE_write_is_sync

Synchronous reporting of device-nGRE write external aborts. 0, asynchronous. 1, synchronous. 2, same as ext_abort_device_write_is_sync.

Type: int. Default value: 0x2.

ext_abort_device_nGRE_write_ras_index

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).

Type: int. Default value: -0x1.

ext_abort_device_nGRE_write_ras_type

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.

Type: int. Default value: -0x1.

ext_abort_device_prefetch_ras_index

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).

Type: int. Default value: -0x1.

ext_abort_device_prefetch_ras_type

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.

Type: int. Default value: -0x1.

ext_abort_device_read_acquire_is_sync

Synchronous reporting of device read with acquire external aborts.

Type: bool. Default value: 0x0.

ext_abort_device_read_is_critical

Critical reporting of device-nGnRE read external aborts.

Type: bool. Default value: 0x0.

ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: bool. Default value: 0x1.

ext_abort_device_read_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number_of_error_records).

Type: int. Default value: 0x0.

ext_abort_device_read_ras_type

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.

Type: int. Default value: 0x0.

ext_abort_device_write_is_critical

Critical reporting of device-nGnRE write external aborts.

Type: bool. Default value: 0x0.

ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: bool. Default value: 0x0.

ext_abort_device_write_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number_of_error_records).

Type: int. Default value: 0x0.

ext_abort_device_write_ras_type

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.

Type: `int`. Default value: `0x0`.

`ext_abort_fill_data`

Returned data, if external aborts are asynchronous.

Type: `int`. Default value: `-0x202020303020203`.

`ext_abort_normal_cacheable_read_is_critical`

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

Type: `bool`. Default value: `0x0`.

`ext_abort_normal_cacheable_read_is_sync`

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

Type: `bool`. Default value: `0x1`.

`ext_abort_normal_cacheable_read_ras_index`

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range `[0, number_of_error_records)`.

Type: `int`. Default value: `0x0`.

`ext_abort_normal_cacheable_read_ras_type`

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.

Type: `int`. Default value: `0x0`.

`ext_abort_normal_cacheable_write_is_critical`

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

Type: `bool`. Default value: `0x0`.

`ext_abort_normal_cacheable_write_is_sync`

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

Type: `bool`. Default value: `0x0`.

`ext_abort_normal_cacheable_write_ras_index`

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range `[0, number_of_error_records)`.

Type: `int`. Default value: `0x0`.

`ext_abort_normal_cacheable_write_ras_type`

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.

Type: `int`. Default value: `0x0`.

`ext_abort_normal_noncacheable_prefetch_ras_index`

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)`.

Type: `int`. Default value: `-0x1`.

`ext_abort_normal_noncacheable_prefetch_ras_type`

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.

Type: `int`. Default value: `-0x1`.

ext_abort_normal_noncacheable_read_is_critical

Critical reporting of normal noncacheable-read external aborts.

Type: `boo1`. Default value: `0x0`.

ext_abort_normal_noncacheable_read_is_sync

Synchronous reporting of normal noncacheable-read external aborts.

Type: `boo1`. Default value: `0x1`.

ext_abort_normal_noncacheable_read_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number_of_error_records).

Type: `int`. Default value: `0x0`.

ext_abort_normal_noncacheable_read_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_normal_noncacheable_write_is_critical

Critical reporting of normal noncacheable write external aborts.

Type: `boo1`. Default value: `0x0`.

ext_abort_normal_noncacheable_write_is_sync

Synchronous reporting of normal noncacheable write external aborts.

Type: `boo1`. Default value: `0x0`.

ext_abort_normal_noncacheable_write_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number_of_error_records).

Type: `int`. Default value: `0x0`.

ext_abort_normal_noncacheable_write_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_normal_wt_cacheable_prefetch_ras_index

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).

Type: `int`. Default value: `-0x1`.

ext_abort_normal_wt_cacheable_prefetch_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_normal_wt_cacheable_read_is_critical

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

Type: `boo1`. Default value: `0x0`.

ext_abort_normal_wt_cacheable_read_is_sync

Synchronous reporting of normal write-through read external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_normal_cacheable_read_is_sync`.

Type: `int`. Default value: `0x2`.

`ext_abort_normal_wt_cacheable_read_ras_index`

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`).

Type: `int`. Default value: `-0x1`.

`ext_abort_normal_wt_cacheable_read_ras_type`

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.

Type: `int`. Default value: `-0x1`.

`ext_abort_normal_wt_cacheable_write_is_critical`

Critical reporting of normal write-through write external aborts.

Type: `bool`. Default value: `0x0`.

`ext_abort_normal_wt_cacheable_write_is_sync`

Synchronous reporting of normal write-through write external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_normal_cacheable_write_is_sync`.

Type: `int`. Default value: `0x2`.

`ext_abort_normal_wt_cacheable_write_ras_index`

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`).

Type: `int`. Default value: `-0x1`.

`ext_abort_normal_wt_cacheable_write_ras_type`

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.

Type: `int`. Default value: `-0x1`.

`ext_abort_prefetch_device_GRE_read_is_critical`

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

Type: `bool`. Default value: `0x0`.

`ext_abort_prefetch_device_GRE_read_is_sync`

Behaviour of external aborts generated by device-GRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `int`. Default value: `0x2`.

`ext_abort_prefetch_device_nGRE_read_is_critical`

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

Type: `bool`. Default value: `0x0`.

`ext_abort_prefetch_device_nGRE_read_is_sync`

Behaviour of external aborts generated by device-nGRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `int`. Default value: `0x2`.

ext_abort_prefetch_device_read_is_critical

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

Type: `bool`. Default value: `0x0`.

ext_abort_prefetch_device_read_is_sync

Behaviour of external aborts generated by device-nGnRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `int`. Default value: `0x2`.

ext_abort_prefetch_is_critical

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

Type: `bool`. Default value: `0x0`.

ext_abort_prefetch_is_sync

Behaviour of external aborts generated by normal writeback cacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort.

Type: `bool`. Default value: `0x1`.

ext_abort_prefetch_noncacheable_read_is_critical

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

Type: `bool`. Default value: `0x0`.

ext_abort_prefetch_noncacheable_read_is_sync

Behaviour of external aborts generated by normal noncacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `int`. Default value: `0x2`.

ext_abort_prefetch_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, `number_of_error_records`).

Type: `int`. Default value: `0x0`.

ext_abort_prefetch_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_prefetch_so_read_is_critical

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

Type: `bool`. Default value: `0x0`.

ext_abort_prefetch_so_read_is_sync

Behaviour of external aborts generated by device=nGnRnE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `int`. Default value: `0x2`.

ext_abort_prefetch_wt_cacheable_read_is_critical

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

Type: `bool`. Default value: `0x0`.

ext_abort_prefetch_wt_cacheable_read_is_sync

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.
Type: `int`. Default value: `0x2`.

ext_abort_so_prefetch_ras_index

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).
Type: `int`. Default value: `-0x1`.

ext_abort_so_prefetch_ras_type

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.
Type: `int`. Default value: `-0x1`.

ext_abort_so_read_is_critical

Critical reporting of device-nGnRnE read external aborts.
Type: `bool`. Default value: `0x0`.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.
Type: `bool`. Default value: `0x1`.

ext_abort_so_read_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number_of_error_records).
Type: `int`. Default value: `0x0`.

ext_abort_so_read_ras_type

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.
Type: `int`. Default value: `0x0`.

ext_abort_so_write_is_critical

Critical reporting of device-nGnRnE write external aborts.
Type: `bool`. Default value: `0x0`.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.
Type: `bool`. Default value: `0x1`.

ext_abort_so_write_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number_of_error_records).
Type: `int`. Default value: `0x0`.

ext_abort_so_write_ras_type

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.
Type: `int`. Default value: `0x0`.

ext_abort_ttw_cacheable_read_is_critical

Critical reporting of TTW cacheable read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_ttw_cacheable_read_is_sync

Synchronous reporting of TTW cacheable read external aborts.

Type: `bool`. Default value: `0x1`.

ext_abort_ttw_cacheable_read_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range `[0, number_of_error_records)`.

Type: `int`. Default value: `0x0`.

ext_abort_ttw_cacheable_read_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_ttw_noncacheable_read_is_critical

Critical reporting of TTW noncacheable read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_ttw_noncacheable_read_is_sync

Synchronous reporting of TTW noncacheable read external aborts.

Type: `bool`. Default value: `0x1`.

ext_abort_ttw_noncacheable_read_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range `[0, number_of_error_records)`.

Type: `int`. Default value: `0x0`.

ext_abort_ttw_noncacheable_read_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_ttw_wt_cacheable_read_is_critical

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

Type: `bool`. Default value: `0x0`.

ext_abort_ttw_wt_cacheable_read_is_sync

Synchronous reporting of TTW write-through cacheable read external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_ttw_cacheable_read_is_sync`.

Type: `int`. Default value: `0x2`.

ext_abort_ttw_wt_cacheable_read_ras_index

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)`.

Type: `int`. Default value: `-0x1`.

ext_abort_ttw_wt_cacheable_read_ras_type

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.

Type: `int`. Default value: `-0x1`.

fault_on_nT_bit_set

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.

Type: `bool`. Default value: `0x1`.

fault_unalign_to_unsupported_access

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.

Type: `int`. Default value: `0x8`.

fault_unaligned_s1_device_s2_fwb

Whether unaligned access with stage1 Device memory and final memory attribute forced to normal by FWB can generate alignment fault.

Type: `bool`. Default value: `0x0`.

flash_protection_enable_at_reset

Enable flash memory protection after reset.

Type: `bool`. Default value: `0x0`.

force_align_pc

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

Type: `bool`. Default value: `0x0`.

fpcr_short_vector_raz

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

Type: `bool`. Default value: `0x0`.

fpsr_res0_stateful_mask

Mask for stateful bits of FPSR which are RES0.

Type: `int`. Default value: `0x0`.

global_debug_rom.ROMDEVID

Value of Debug Rom Device Identification Register.

Type: `int`. Default value: `0x0`.

global_debug_rom.ROMPIDR

Value of Debug Rom Peripheral Identification Register.

Type: `int`. Default value: `0x4000bb000`.

global_debug_rom.ROMPRIDR0

Value of Debug ROM Power RequestID Register.

Type: `int`. Default value: `0x1`.

hardware_translation_table_update_implemented

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.

Type: `int`. Default value: `0x1`.

has_16bit_asids

Enable 16-bit ASIDs.

Type: `bool`. Default value: `0x1`.

has_16bit_vmids

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.

Type: `int`. Default value: `0x1`.

has_16k_granule

Implement the 16k LPAE translation granule.

Type: `bool`. Default value: `0x0`.

has_4k_granule

Implement the 4k LPAE translation granule.

Type: `bool`. Default value: `0x1`.

has_64k_granule

Implement the 64k LPAE translation granule.

Type: `bool`. Default value: `0x1`.

has_aarch32_dbgdidr_etc

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

Type: `bool`. Default value: `0x1`.

has_aarch64

All implemented exception levels can run in AArch64.

Type: `bool`. Default value: `0x0`.

has_ccidx

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

Type: `bool`. Default value: `0x0`.

has_cluster_l1cache_size

Whether core supports cluster level l1cache size.

Type: `bool`. Default value: `0x1`.

has_coherent_icache

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`. Default value: `0x0`.

has_const_pac

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.

Type: `int`. Default value: `0x0`.

has_cvadp_support

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.

Type: `int`. Default value: `0x0`.

has_debug_rom

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

Type: `bool`. Default value: `0x1`.

has_delayed_ctireg

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

Type: `bool`. Default value: `0x0`.

has_delayed_dbgreg

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

Type: `bool`. Default value: `0x0`.

has_delayed_mdscr_el1

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

Type: `bool`. Default value: `0x0`.

has_delayed_oslar_el1

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

Type: `bool`. Default value: `0x0`.

has_delayed_pmureg

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

Type: `bool`. Default value: `0x0`.

has_delayed_sysreg

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

Type: `bool`. Default value: `0x0`.

has_dgh

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.

Type: `int`. Default value: `0x1`.

has_e0pd

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.

Type: `int`. Default value: `0x0`.

has_enhanced_pac

If pointer authentication is enabled then implement enhanced PAC.

Type: `bool`. Default value: `0x0`.

has_exception_trapping_form_of_vector_catch

Implement the exception trapping form of vector catch debug event.

Type: `bool`. Default value: `0x1`.

has_export_m_port

The interrupt distributor has an optional interrupt export port for routing interrupts to an external device.

Type: `bool`. Default value: `0x1`.

has_far_not_valid

Implements FnV bit in ESR_ELx and xFSR, FAR not valid for synchronous external aborts.

Type: `bool`. Default value: `0x0`.

has_far_not_valid_dfsc

Implements FnV bit in ESR_ELx, FAR not valid for synchronous external aborts for Data Abort.

Type: `bool`. Default value: `0x0`.

has_far_not_valid_ifsc

Implements FnV bit in ESR_ELx and xFSR, FAR not valid for synchronous external aborts for Instruction Abort.

Type: `bool`. Default value: `0x0`.

has_flash

Flash Port present.

Type: `bool`. Default value: `0x0`.

has_flash_protection

Implement flash memory protection.

Type: `bool`. Default value: `0x0`.

has_fpl6

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.

Type: `int`. Default value: `0x1`.

has_generic_authentication

Implement ARMv8.3 generic authentication. Possible values of this parameter are: - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x1`.

has_hardware_translation_table_update

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.

Type: `int`. Default value: `0x2`.

has_internal_gic_iri

Is Internal GIC IRI implemented.

Type: `bool`. Default value: `0x0`.

has_itd

Implement the optional IT disable feature.

Type: `bool`. Default value: `0x1`.

has_large_system_ext

Implement the ARMv8 Large System Extensions (FEAT_LSE).

Type: `bool`. Default value: `0x0`.

has_large_va

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.

Type: `int`. Default value: `0x0`.

has_llpp

Low Latency Peripheral Port present.

Type: `bool`. Default value: `0x0`.

has_no_os_double_lock

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.

Type: `int`. Default value: `0x0`.

has_par_bit10_razwi

Whether PAR_EL1[10] is RAZ/WI.

Type: `bool`. Default value: `0x0`.

has_partial_delayed_mdscr_el1

`has_delayed_oslar_el1` only apply to some bits of MDSCR_EL1 (MDE, KDE, TDCC, SS).

Type: `bool`. Default value: `0x0`.

has_pc_sample_based_profiling

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

Type: `bool`. Default value: `0x1`.

has_pl2

Whether EL2 is implemented.

Type: `bool`. Default value: `0x1`.

has_pmc

Programmable MBIST controllers implemented.

Type: `bool`. Default value: `0x0`.

has_pmu

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.

Type: `int`. Default value: `0x1`.

has_pointer_authentication

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.

Type: `int`. Default value: `0x1`.

has_prediction_invalidation_instructions

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.

Type: `int`. Default value: `0x1`.

has_qarma3_pac

Supports QARMA3 pointer authentication algorithm (FEAT_PACQARMA3).

Type: `bool`. Default value: `0x0`.

has_ras

Implements the ARMv8R-64 RAS Extension. 0 = NO_RAS, 1 = MINIMAL_RAS, 2 = FULL_RAS.

Type: `int`. Default value: `0x0`.

has_ras_armv84_extension

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.

Type: `int`. Default value: `0x0`.

has_ras_double_fault

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.

Type: `int`. Default value: `0x0`.

has_restriction_on_speculative_data_loaded

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.

Type: `int`. Default value: `0x1`.

has_self_hosted_trace_extension

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.

Type: `int`. Default value: `0x1`.

has_small_page_table

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.

Type: `int`. Default value: `0x1`.

has_speculation_barrier_inst

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.

Type: `int`. Default value: `0x1`.

has_spp

Shared Peripheral Port present.

Type: `bool`. Default value: `0x0`.

has_synchronous_load_atomics

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

Type: `bool`. Default value: `0x1`.

has_synchronous_load_atomics_noncacheable

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

Type: `bool`. Default value: `0x1`.

has_synchronous_store_atomics

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

Type: `bool`. Default value: `0x0`.

has_synchronous_store_atomics_noncacheable

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

Type: `bool`. Default value: `0x0`.

has_tlb_conflict_abort

Detected inconsistent TLB content generate aborts.

Type: `bool`. Default value: `0x0`.

has_tlb_pa_caching

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_nTLBPAA).

Type: `bool`. Default value: `0x0`.

has_unsupported_exclusive_fault

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

Type: `bool`. Default value: `0x1`.

has_v8_4_pmu_extension

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.

Type: `int`. Default value: `0x1`.

has_v8_5_debug_over_power_down

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.

Type: `int`. Default value: `0x0`.

has_v8_6_pmu_events

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.

Type: `int`. Default value: `0x1`.

has_writebuffer

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

Type: `bool`. Default value: `0x0`.

hcptr_tta_behaviour

Behaviour of HCPTTR.TTA when there is no CP14 ETM interface. 0, RAZ/WI. 1, RAO/WI. 2, stateful.

Type: `int`. Default value: `0x2`.

hcr_swio_res1

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

Type: `bool`. Default value: `0x0`.

hsr_uncond_cc

Condition codes reported in HSR as AL if it passes.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-log2linelen

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.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-nprefetch

Number of next sequential instruction cache lines to prefetch. This is only used when `icache-prefetch_enabled=true`.

Type: `int`. Default value: `0x1`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-prefetch_level

0 based cache level at which instructions are pre-fetched. This is only used when `icache-prefetch_enabled=true`.

Type: `int`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_bus_width_in_bytes

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

Type: `int`. Default value: `0x8`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

icache-ways

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

Type: `int`. Default value: `0x2`.

imp_def_functionality_behaviour

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

Type: `int`. Default value: `0x0`.

instruction_tlb_size

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

Type: `int`. Default value: `0x0`.

is_debug_state_pmu_snapshot_allowed

If true, PMU snapshot is allowed in debug state.

Type: `bool`. Default value: `0x1`.

is_first_pcsr_sample_ignored

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

Type: `bool`. Default value: `0x0`.

is_uniprocessor

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

Type: `bool`. Default value: `0x0`.

ish_is_osh

Whether Innershareable is same as OuterShareable.

Type: `bool`. Default value: `0x0`.

itd_conditional_instructions_are_32bit

When SCTLR_ELx.ITD=1, an IT instruction plus a T16 instruction are considered a single 32bit conditional instruction.

Type: `bool`. Default value: `0x0`.

l1cache_has_r52_cache_policy

Whether l1cache has r52 cache policy.

Type: `bool`. Default value: `0x0`.

l1cache_has_rsvd_flash_ways

Whether l1 cache segregates ways for flash and AXI data.

Type: `bool`. Default value: `0x0`.

l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-read_bus_width_in_bytes

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

Type: `int`. Default value: `0x8`.

l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-ways

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

Type: `int`. Default value: `0x10`.

l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-write_bus_width_in_bytes

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

Type: `int`. Default value: `0x8`.

l2cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

memory.ext_slave_base

Base address of Slave Port. Each core's region will offset by `ext_slave_size_per_core`.

Type: `int`. Default value: `0x0`.

memory.ext_slave_size_per_core

Size of Slave region for each core.

Type: `int`. Default value: `0x0`.

memory.flash_base

Base address of Flash.

Type: `int`. Default value: `0x0`.

memory.flash_size

Size of the Flash RAM.

Type: `int`. Default value: `0x0`.

memory.has_llram

Low-Latency RAM present.

Type: `bool`. Default value: `0x0`.

memory.l2_cache.is_inner_cacheable

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

Type: `bool`. Default value: `0x1`.

memory.l2_cache.is_inner_shareable

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

Type: `bool`. Default value: `0x1`.

memory.llram_base

Base address of LLRAM.

Type: `int`. Default value: `0x0`.

memory.llram_enable_at_reset

Whether llram is enabled at reset.

Type: `bool`. Default value: `0x0`.

memory.llram_shared

Controls the Low-Latency RAM's sharability attribute.

Type: `bool`. Default value: `0x0`.

memory.llram_size

Size of the LLRAM.

Type: `int`. Default value: `0x0`.

memory.scu_present

L1 Caches are coherent.

Type: `bool`. Default value: `0x1`.

memory.transmit_vmid_in_user_flags

Transmit VMID in transaction attributes.

Type: `bool`. Default value: `0x0`.

MIDR

Value of MIDR_EL1 register.

Type: `int`. Default value: `0x410fd0f0`.

mixed_endian

Implement support for mixed endianness. 0, not supported. 1, supported at all exception levels. 2, supported at ELO only.

Type: `int`. Default value: `0x1`.

mpidr_layout

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

Type: `int`. Default value: `0x0`.

non_secure_vgic_alias_when_ns_only

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).

Type: `int`. Default value: `0x0`.

NUM_CORES

Number of cores in cluster.

Type: `int`. Default value: `0x1`.

num_protection_regions_sl

Number of v8-R protection regions.

Type: `int`. Default value: `0x20`.

num_protection_regions_s2

Number of v8-R hyp protection regions.

Type: `int`. Default value: `0x20`.

num_spi

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

Type: `int`. Default value: `0x20`.

number_of_error_records

Cores Number of Error records supported for RAS.

Type: `int`. Default value: `0x0`.

PA_SIZE

Physical address range supported (FEAT_LPA).

Type: `int`. Default value: `0x28`.

page_based_hardware_attributes

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).

Type: `int`. Default value: `0x0`.

per_core_master_supported

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

Type: `bool`. Default value: `0x0`.

pfr1_csv2_frac

Fractional revision number ID_AA64PFR1_EL1.CSV2_frac when ID_AA64PFR0_EL1.CSV==1 for CSV2 extension (FEAT_CSV2_1p1, FEAT_CSV2_1p2).

Type: `int`. Default value: `0x0`.

pmb_idr_external_abort

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.

Type: `int`. Default value: `0x0`.

pmb_idr_flag_updates

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

Type: `bool`. Default value: `0x1`.

PMCEID0

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

Type: `int`. Default value: `0xfffffffffff`.

PMCEID1

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

Type: `int`. Default value: `0xfffffffffff`.

pmmir_ell_bus_slots

Largest value by which BUS_ACCESS can increment over BUS_CYCLES cycles.

Type: `int`. Default value: `0x0`.

pmmir_ell_bus_width

Width, in bytes, of accesses counted by BUS_ACCESS.

Type: `int`. Default value: `0x0`.

pms_idr_max_size

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

Type: `int`. Default value: `0x6`.

PMSIDR.ArchInst

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.

Type: `bool`. Default value: `0x1`.

PMSIDR.CRR

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

Type: `bool`. Default value: `0x0`.

PMSIDR.LDS

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.

Type: `bool`. Default value: `0x0`.

pmu-num_counters

Number of PMU counters implemented.

Type: `int`. Default value: `0x8`.

pmu_cycle_counter_counts_actual_cycles

If true and Timing annotation is enabled, PMU cycle counter counts actual cycles, otherwise counts instructions executed.

Type: `bool`. Default value: `0x0`.

pmu_has_chain_event

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

Type: `bool`. Default value: `0x1`.

PMUPIDR

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

Type: `int`. Default value: `0x0`.

pseudo_fault_generation_feature_register

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,

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.

Type: `string`. Default value: `""`.

pstate_ssbs_type

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).

Type: `int`. Default value: `0x0`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

ram_protection_enable_at_reset

Enable TCM, L1Cache memory protection after reset.

Type: `bool`. Default value: `0x0`.

randomize_unknowns_at_reset

Will fill in unknown bits in registers at reset with random value using register_reset_data as seed, it overrides scramble_unknowns_at_reset.

Type: `bool`. Default value: `0x0`.

ras_extra_configurations

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}].

Type: `string`. Default value: `""`.

ras_pfg_clock_mhz

RAS Pseudo-Fault generation clock rate in MHz.

Type: `int`. Default value: `0x18`.

register_reset_data

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

Type: `int`. Default value: `0x0`.

register_reset_data_hi

Data used to fill the upper-half of 128-bit registers when the bits become UNKNOWN at reset.

Type: `int`. Default value: `0x0`.

report_iside_cmo_ifsr

fault info for an iside cache maintenance operation is reported in the IFSR.

Type: `bool`. Default value: `0x1`.

reserved_HMC_SSC_PAC_treated_disabled

When DBG[B|W]CR.{HMC,SSC,PAC} bits configuration is reserved, this parameter controls whether breakpoints/watchpoints are treated as Disabled or not.

Type: `bool`. Default value: `0x0`.

restriction_on_speculative_execution

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).

Type: `int`. Default value: `0x0`.

rmr_always_implemented

Always implement RMR_ELx, RMR, or HRMR at the highest implemented exception level, even if that exception level cannot use both AArch32 and AArch64.

Type: `bool`. Default value: `0x0`.

scheduler_mode

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!.

Type: `int`. Default value: `0x0`.

scr_nET_writeable

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

Type: `bool`. Default value: `0x0`.

scramble_unknowns_at_reset

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

Type: `bool`. Default value: `0x1`.

spp.base

Sets the base address of Shared Peripheral Port.

Type: `int`. Default value: `0x0`.

spp.size

Sets the size of SPP(in bytes).

Type: `int`. Default value: `0x1000`.

spsr_el3_is_mapped_to_spsr_mon

Whether SPSR_EL3 is mapped to AArch32 register SPSR_mon.

Type: `bool`. Default value: `0x0`.

spsr_m4_res0

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

Type: `bool`. Default value: `0x0`.

stage12_tlb_size

If VMSA is supported at stage1, number of stage1+2 tlb entries. If instruction_tlb_size !=0, this is treated as dtlb size.

Type: int. Default value: 0x0.

stage1_tlb_size

Number of stage1 only tlb entries.

Type: int. Default value: 0x0.

stage1_walkcache_size

Number of stage1 only walk cache entries.

Type: int. Default value: 0x0.

supports_multi_threading

Sets the MPIDR.MT bit. Setting this to true hints the the cluster is multi-threading compatible.

Type: bool. Default value: 0x0.

take_ccfail_tsc_trap

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.

Type: bool. Default value: 0x0.

take_ccfail_undef

UNDEF exception is taken even if condition code check fails.

Type: bool. Default value: 0x1.

tidcp_traps_el0_undef_imp_def

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

Type: bool. Default value: 0x1.

tlb_latency

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

Type: int. Default value: 0x0.

trace_icc_registers_as_icv_when_redirected

If true, update trace with ICV, instead of ICC when ICV registers are accessed depending on the core state.

Type: bool. Default value: 0x0.

trace_physical_registers_when_host_virtualisation_enabled

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.

Type: int. Default value: 0x0.

trap_dc_cmo_to_pou_if_nop

Whether traps to DC CMO operations to PoU are ignored if the same is treated as NOP.

Type: bool. Default value: 0x1.

trap_ic_cmo_to_pou_if_nop

Whether traps to IC CMO operations to PoU are ignored if the same is treated as NOP.

Type: `bool`. Default value: `0x1`.

trap_reserved_group3_id_regs

Whether setting HCR_EL2.TID3 traps reserved group3 id registers.

Type: `bool`. Default value: `0x0`.

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

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.

Type: `int`. Default value: `0x0`.

treat-dcache-cmos-to-pou-as-nop

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.

Type: `int`. Default value: `0x0`.

treat-dcache-invalidate-as-clean-invalidate

Treat data cache invalidate operations as clean and invalidate.

Type: `bool`. Default value: `0x0`.

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

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.

Type: `int`. Default value: `0x0`.

treat_forced_normal_as_device_for_excl_atomics

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.

Type: `bool`. Default value: `0x0`.

treat_pld_as_nop

If true, treat PLD as NOP.

Type: `bool`. Default value: `0x0`.

treat_pli_as_nop

If true, treat PLI as NOP.

Type: `bool`. Default value: `0x0`.

treat_wfi_wfe_as_nop

If true, never go into wait state for WFI or WFE instructions.

Type: `bool`. Default value: `0x0`.

truncate_pc_on_illegal_exception_return_to_aarch32

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

Type: `bool`. Default value: `0x1`.

unification-level

Level of Unification Inner Shareable for the cache hierarchy.

Type: `int`. Default value: `0x1`.

unification-uniprocessor-level

Level of Unification Uniprocessor for the cache hierarchy.

Type: `int`. Default value: `0x1`.

unpred_edscr_rw_unknown_bits_read_as_1

Unknown(x) bits in RW field in EDSCR are read as 1 instead of 0.

Type: `bool`. Default value: `0x0`.

unpred_edscr_status_read_as_no_syndrome

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.

Type: `bool`. Default value: `0x0`.

unpred_mrsmsr_currentlymapped_undef

UNPREDICTABLE register access (accessible from current mode using different instruction) modeled as NOP when false and UNDEF when true.

Type: `bool`. Default value: `0x0`.

unpred_mrsmsr_protfailed_undef

UNPREDICTABLE register access (not accessible from current PL and security state) modeled as NOP when false and UNDEF when true.

Type: `bool`. Default value: `0x0`.

unpred_stage2_mpu_and_bg_disabled

Constrained unpredictable when stage2 MPU and background disabled. 0, Stage-2 level 0 translation fault(Default). 1, Unknown memory attributes.

Type: `int`. Default value: `0x0`.

unpred_tsize_aborts

Behaviour when TSize is out of range. 0, force into range. 1, translation fault, forces `unpred_tsize_pamax_aborts` to 1.

Type: `bool`. Default value: `0x0`.

unpred_tsize_pamax_aborts

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.

Type: `bool`. Default value: `0x0`.

unpredictable_exclusive_abort_memtype

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).

Type: `int`. Default value: `0x0`.

unpredictable_hvc_behaviour

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

Type: `int`. Default value: `0x0`.

unpredictable_smc_behaviour

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

Type: `int`. Default value: `0x0`.

unsupported_atomic_fault_type

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.

Type: `int`. Default value: `0x0`.

unsupported_hw_update_fault_type

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).

Type: `int`. Default value: `0x0`.

use_tlb_contig_hint

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

Type: `bool`. Default value: `0x0`.

VMSA_supported

VMSA is supported at EL1.

Type: `bool`. Default value: `0x1`.

vpu_datapath_width

VPU data path width.

Type: `int`. Default value: `0x80`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

warn_unpredictable_in_v7

If true, behaviour which is unpredictable in V7 yet is predictable in V8 will produce a warning.

Type: `bool`. Default value: `0x1`.

watchpoint-log2secondary_restriction

log2 size of secondary restriction of FAR/EDWAR possible values on watchpoint hit for load/store operations.

Type: `int`. Default value: `0x0`.

wfe_wakeup_delay

Configure WFE wakeup delay in CPU cycles.

Type: `int`. Default value: `0x0`.

wfi_wakeup_delay

Configure WFI wakeup delay in CPU cycles.

Type: `int`. Default value: `0x0`.

4.5.2 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 4-75: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

AEMvACT contains the following CADI targets:

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

AEMvACT contains the following MTI components:

- [ARM_AEM-A_MP](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Ports added:

- `hacdbirq`

Parameters added:

- `$<ras_dependent_feature_params>`
- `dsb_accumulate_threshold`
- `eddfrr1_reads_idreg_mask`
- `eddfrr2_reads_idreg_mask`
- `eddfrr_reads_idreg_mask`
- `fpsr_res0_stateful_mask`
- `has_ras_delegated_serror_exceptions_for_el3`
- `hdbss_error_fault_type`
- `l3cache-mpamf.arch_major_ver`

- `l3cache-mpamf.arch_minor_ver`
- `mops_cpy_pre_size_threshold`
- `mops_inst_cpy_zero_size_can_fault`
- `mops_inst_set_zero_size_can_fault`
- `pacm_support_level`
- `report_second_access_align_fault_non_atomic_pair_access`
- `report_second_access_mmu_fault_non_atomic_pair_access`
- `spmu_support_level`
- `store_excl_fail_tag_check_action`
- `use_architectural_names`

Parameters removed:

- `DCRPZID_log2_block_size`
- `dc_ops_remove_poison`
- `has_ras_rpz`
- `has_spmu`
- `poison_range_end_addr`
- `poison_range_start_addr`

About AEMvACT

AEMvACT implements all architectural features in Arm®v8-A and Arm®v9-A.

This model 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 Arm®v8.3-A onwards. Support is at EAC quality unless otherwise stated.

Table 4-76: Architectural features implemented in Fast Models

Architecture version	Feature	Fast Models version									
		11.24	11.23	11.22	11.21	11.20	11.19	11.18	11.17	11.16	11.15
Arm®v8.3-A	All. For details, see list after table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Arm®v8.4-A	All. For details, see list after table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Arm®v8.5-A	Memory Tagging Extension (MTE) functionality	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	MTE performance	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	All except MTE. For details, see list after table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Arm®v8.6-A	All. For details, see list after table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Arm®v8.7-A	Limited TLBI maintenance	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	All except TLBI maintenance. For details, see list after table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Arm®v8.8-A	All	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
Arm®v8.9-A	Guarded Call Stack, 128-bit page table descriptors, and PMU updates	✓	✓	✓	✓	-	-	-	-	-	-
	All other features	✓	✓	✓	✓	✓	-	-	-	-	-
Arm®v9.0-A to Arm®v9.3-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).	✓	✓	✓	✓	✓	✓	-	-	-	-

Architecture version	Feature	Fast Models version									
		11.24	11.23	11.22	11.21	11.20	11.19	11.18	11.17	11.16	11.15
	All other features. For details, see the list below this table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Arm®v9.4-A	RME without Secure EL2.	✓	✓	✓	✓	-	-	-	-	-	-
	All other features.	✓	✓	✓	✓	✓	-	-	-	-	-
Arm®v9.5-A	All features.	✓	-	-	-	-	-	-	-	-	-

- Arm®v8.3-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
- Arm®v8.4-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
- Arm®v8.5-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
- Arm®v8.6-A features implemented include:
 - General Matrix Multiply (GEMM)
 - EnhancedPAC2
 - FPAC
 - Data Gathering Hint
 - Additional PMU events
 - Enhanced Counter Virtualization
- Arm®v8.7-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 Arm®v9-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 4-77: 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.
CNTHVSIRQ[8]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[8]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[8]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[8]	Signal	Master	Timer signals to SOC.
commirq[8]	Signal	Master	Interrupt signal from debug communications channel.

Name	Protocol	Type	Description
config64[8]	Signal	Slave	Register width after reset.
cp15sdisable[8]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[8]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
CRITICALIRQ[8]	Signal	Master	RAS Critical Error Interrupt.
cryptodisable[8]	Signal	Slave	Disable cryptography extensions after reset.
cti[8]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
cti0extin[4]	Signal	Slave	CTI trace inputs for core 0.
cti0extout[4]	Signal	Master	CTI trace outputs for core 0.
cti1extin[4]	Signal	Slave	CTI trace inputs for core 1.
cti1extout[4]	Signal	Master	CTI trace outputs for core 1.
cti2extin[4]	Signal	Slave	CTI trace inputs for core 2.
cti2extout[4]	Signal	Master	CTI trace outputs for core 2.
cti3extin[4]	Signal	Slave	CTI trace inputs for core 3.
cti3extout[4]	Signal	Master	CTI trace outputs for core 3.
cti4extin[4]	Signal	Slave	CTI trace inputs for core 4.
cti4extout[4]	Signal	Master	CTI trace outputs for core 4.
cti5extin[4]	Signal	Slave	CTI trace inputs for core 5.
cti5extout[4]	Signal	Master	CTI trace outputs for core 5.
cti6extin[4]	Signal	Slave	CTI trace inputs for core 6.
cti6extout[4]	Signal	Master	CTI trace outputs for core 6.
cti7extin[4]	Signal	Slave	CTI trace inputs for core 7.
cti7extout[4]	Signal	Master	CTI trace outputs for core 7.
ctidbgirq[8]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen[8]	Signal	Slave	External debug interface.
dbgnopwrdown[8]	Signal	Master	Debug no power down request.
dbgpwrdownack[8]	Signal	Master	Debug power down acknowledge.
dbgpwrdownreq[8]	Signal	Slave	Debug power down request.
dbgpwrupreq[8]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
ERRORIRQ[8]	Signal	Master	RAS Error Recovery Interrupt.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
external_trace_reset[8]	Signal	Slave	ETMv4 External Trace Reset signal.
FAULTIRQ[8]	Signal	Master	RAS Fault Handling Interrupt
fiq[8]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
fiq_nmi[8]	Signal	Slave	-

Name	Protocol	Type	Description
gicv3_redistributor_s[8]	GICv3Comms	Slave	GICv3 cpu interface ports.
hacdbsir[8]	Signal	Master	Interrupt signal from the HACDBS unit.
irq[8]	Signal	Slave	This signal drives the CPUs interrupt handling.
irq_nmi[8]	Signal	Slave	-
irqs[224]	Signal	Slave	These signals drive the CPU's interrupt controller interrupt lines.
l2reset	Signal	Slave	This signal resets timer and interrupt controller.
memorymapped_amu_s	PVBus	Slave	External interface for amu.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[8]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region.
pmbsir[8]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuir[8]	Signal	Master	Interrupt signal from performance monitoring unit.
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.
standbyfil2	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.

Name	Protocol	Type	Description
ticks[8]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trace_unit_reset[8]	Signal	Slave	ETMv4 Trace Unit Reset signal.
trbirq[8]	Signal	Master	Interrupt signal from the trace buffer unit.
vcpumntirq[8]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[8]	Signal	Slave	Virtual FIQ input. Note that the irq/fiq pins are wired directly to the core if there is no internal VGIC. If there is an internal VGIC then these are ignored.
vinithi[8]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[8]	Signal	Slave	Virtual IRQ input. Note that the irq/fiq pins are wired directly to the core if there is no internal VGIC. If there is an internal VGIC then these are ignored.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[8]	Signal	Slave	Processor Virtual System Error Interrupt request.

Parameters for ARM_AEM-A_MP

cpu0.aarch32_reset_from_impdef_addr

If PE resets into AArch32, Whether execution starts from IMPDEF address or hi/low vector.
Type: bool. Default value: 0x1.

cpu0.ase-present

Set whether the model has been built with NEON support.
Type: bool. Default value: 0x1.

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.clock_divider

Clock divider ratio for asymmetric MP clocking.
Type: int. Default value: 0x1.

cpu0.clock_multiplier

Clock divider ratio for asymmetric MP clocking.
Type: int. Default value: 0x1.

cpu0.CONFIG64

Register width configuration at reset. 0, AArch32. 1, AArch64.
Type: bool. Default value: 0x1.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: `bool`. Default value: `0x0`.

cpu0.CP15SDISABLE2

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

Type: `bool`. Default value: `0x0`.

cpu0.crypto_aes

AES instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 2, AES and PMULL instructions implemented (FEAT_AES, FEAT_PMULL).

Type: `int`. Default value: `0x2`.

cpu0.crypto_sha1

SHA-1 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA1 instructions implemented (FEAT_SHA1).

Type: `int`. Default value: `0x1`.

cpu0.crypto_sha256

SHA-256 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA256 instructions implemented (FEAT_SHA256).

Type: `int`. Default value: `0x1`.

cpu0.crypto_sha3

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.

Type: `int`. Default value: `0x0`.

cpu0.crypto_sha512

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.

Type: `int`. Default value: `0x0`.

cpu0.crypto_sm3

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.

Type: `int`. Default value: `0x0`.

cpu0.crypto_sm4

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.

Type: `int`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.cti-intack_mask

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

Type: `int`. Default value: `0x1`.

cpu0.cti-number_of_claim_bits

Number of implemented bits in CTICLAIMSET.

Type: `int`. Default value: `0x0`.

cpu0.cti-number_of_triggers

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

Type: `int`. Default value: `0x8`.

cpu0.DCZID-log2-block-size

Log2 of the block size cleared by DC ZVA instruction (as read from DCZID_EL0).

Type: `int`. Default value: `0x8`.

cpu0.DCZVA_single_write

Execute the DCZVA as a single write.

Type: `bool`. Default value: `0x0`.

cpu0.enable_crc32

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

Type: `int`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.etm-present

Set whether the model has ETM support.

Type: `bool`. Default value: `0x1`.

cpu0.force-fpsid

Override the FPSID value.

Type: `bool`. Default value: `0x0`.

cpu0.force-fpsid-value

Value to override the FPSID value to.

Type: `int`. Default value: `0x0`.

cpu0.has_hcptr_tase

If false, HCPTR.TASE is RES0.

Type: `bool`. Default value: `0x1`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

`cpu0.MPIDR-override`

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

Type: `int`. Default value: `0x0`.

`cpu0.number-of-breakpoints`

Number of breakpoints.

Type: `int`. Default value: `0x10`.

`cpu0.number-of-context-breakpoints`

Number of breakpoints that are context aware.

Type: `int`. Default value: `0x10`.

`cpu0.number-of-watchpoints`

Number of watchpoints.

Type: `int`. Default value: `0x10`.

`cpu0.operation_bandwidth`

Operation width for ARMv8.4 PMU extension.

Type: `int`. Default value: `0x1`.

`cpu0.RVBAR`

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

`cpu0.RVBAR32`

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

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-prefix

Prefix semihosting output with target instance name.
Type: `bool`. Default value: `0x0`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stderr_istty

Result for semihost istry call when argument is stderr.
Type: `bool`. Default value: `0x1`.

cpu0.semihosting-stdin_istty

Result for semihost istry call when argument is stdin.
Type: `bool`. Default value: `0x1`.

cpu0.semihosting-stdout_istty

Result for semihost istry call when argument is stdout.
Type: `bool`. Default value: `0x1`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: `int`. Default value: `0xab`.

cpu0.semihosting-use_stderr

Send stderr from the simulated process to host stderr.
Type: `bool`. Default value: `0x0`.

cpu0.SMPnAMP

Enable broadcast messages necessary for correct SMP operation at reset.
Type: `bool`. Default value: `0x1`.

cpu0.TEINIT

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: `bool`. Default value: `0x0`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF enable_trace_special_hlt_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: int. Default value: 0xf000.

cpu0.unpredictable_WPMASKANDBAS

Constrained unpredictable handling of watchpoints when mask and BAS fields specified. 0, IGNOREMASK. 1, IGNOREBAS (default). 2, REPEATBAS8. 3, REPEATBAS.

Type: int. Default value: 0x1.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool. Default value: 0x0.

cpu0.vfp-present

Set whether the model has VFP support.

Type: bool. Default value: 0x1.

cpu0.vfp-traps

Implement support for trapping floating-point exceptions.

Type: bool. Default value: 0x1.

cpu0.vfp-traps-show-all

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

Type: bool. Default value: 0x0.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: bool. Default value: 0x0.

cpu0.wfet_early_or_delayed_timeout

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

Type: int. Default value: 0x0.

cpu0.wfit_early_or_delayed_timeout

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

Type: int. Default value: 0x0.

Parameters for Cluster_ARM_AEM-A_MP**\$<ras_dependent_feature_params>**

Dependent param generation for RAS based on SMCU count.

Type: string. Default value: "".

advsimd_bf16_support_level

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.

Type: int. Default value: 0x0.

advsimd_i8mm_support_level

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).
Type: `int`. Default value: `0x0`.

amair_reg_rw_mask

RW mask for implementation-defined registers.
Type: `int`. Default value: `0x0`.

amu_aux_voffset_mask

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.
Type: `int`. Default value: `0x0`.

amu_has_external_interface

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.
Type: `int`. Default value: `0x0`.

amu_has_sysreg_interface

Implement system register access to activity monitor unit from ARMv8.4.
Type: `bool`. Default value: `0x1`.

amu_reset_domain

Reset domain for activity monitor unit. 0, COLD_RESET. 1, WARM_RESET. 2, NONE.
Type: `int`. Default value: `0x0`.

amu_version

Selects the activity monitor version implemented - 1, AMUv1 for Armv8.4 is implemented. - 2, AMUv1 for Armv8.6 is implemented (FEAT_AMUv1p1).
Type: `int`. Default value: `0x1`.

arm_v8_7_accelerator_support_level

Implements Armv8.7 accelerator support instructions LD/ST64B, ST64BV, ST64BV0, etc. (FEAT_LS64, FEAT_LS64_V, FEAT_LS64_ACCDATA).
Type: `int`. Default value: `0x0`.

changing_block_size_without_bbm_support

Level of support for changing block size without break-before-make (FEAT_BBM).
Type: `int`. Default value: `0x0`.

checked_pointer_arithmetic_support_level

Specify the Checked Pointer Arithmetic support level: 0, not implemented. 1, FEAT_CPA is implemented. 2, FEAT_CPA2 is implemented.
Type: `int`. Default value: `0x0`.

clear_ec_in_debug_state

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.

Type: `bool`. Default value: `0x0`.

clear_IT_when_IL_set

Clear IT bits when performing a *legal* exception return to AArch32 when IL is set.

Type: `bool`. Default value: `0x0`.

clear_IT_when_IL_set_explicitly

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.

Type: `bool`. Default value: `0x0`.

core_cache_protection

`core_cache_protection` can change `ERROFR`, `ERROPFGF` and `ERROPFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int`. Default value: `-0x1`.

cpy_mops_option

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.

Type: `int`. Default value: `0x0`.

cpyf_mops_option

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.

Type: `int`. Default value: `0x0`.

dczva_reports_lowest_addr_on_tag_check_fail

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

Type: `bool`. Default value: `0x0`.

dczva_wp_far_behaviour

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.

Type: `int`. Default value: `0x0`.

delayed_dbgreg_between_secure_views

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.

Type: `int`. Default value: `0x1`.

delayed_pmureg_between_secure_views

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.

Type: `int`. Default value: `0x1`.

disable_sve_plugin

If true, SVE will not be implemented in this processor even if the plugin is loaded (FEAT_SVE).

Type: `bool`. Default value: `0x0`.

early_implicit_error_sync_event_behaviour

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`.

Type: `int`. Default value: `0x0`.

ecv_support_level

Implement Enhanced Counter Virtualization feature from ARMv8.6. `0`, Not supported. `1`, fully supported without `CNTPOFF`. `2`, fully supported with `CNTPOFF` (FEAT_ECV).

Type: `int`. Default value: `0x0`.

eddfr1_reads_idreg_mask

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.

Type: `int`. Default value: `-0x1`.

eddfr2_reads_idreg_mask

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.

Type: `int`. Default value: `-0x1`.

eddfr_reads_idreg_mask

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.

Type: `int`. Default value: `-0x1`.

edxfr_reads_idreg

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 'eddfr*_reads_idreg_mask' parameter.

Type: `bool`. Default value: `0x0`.

e10_e11_only_non_secure

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

Type: `bool`. Default value: `0x0`.

enhanced_pac2_level

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).

Type: `int`. Default value: `0x0`.

error_record_feature_register

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,"INJ":0x0,"FRX":0x0,"UC":0x0,"UEU":0x0,"UER":0x0,"UEO":0x0,"DE":0x0,"CI":0x0,"TS":0x0,"Visibility":0x0,"Core":0x0,"Cluster":0x0}]
```

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".

Type: `string`. Default value: `""`.

error_record_feature_register_json_file

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

Type: `string`. Default value: `""`.

ERXMISC0_mask

Write Mask for ERXMISC0 RAS Register.

Type: `int`. Default value: `0x0`.

esr_write_update_res0

If true, and RASv2 is enabled, then ESR_ELx.WU field is RES0 for errors on both loads and stores (FEAT_RASv2).

Type: `bool`. Default value: `0x0`.

ets_level

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.

Type: `int`. Default value: `0x0`.

ext_abort_device_GRE_prefetch_ras_index

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`).

Type: `int`. Default value: `-0x1`.

ext_abort_device_GRE_prefetch_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_device_GRE_read_ras_index

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`).

Type: `int`. Default value: `-0x1`.

ext_abort_device_GRE_read_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_device_GRE_write_ras_index

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`).

Type: `int`. Default value: `-0x1`.

ext_abort_device_GRE_write_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_device_nGRE_prefetch_ras_index

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`).

Type: `int`. Default value: `-0x1`.

ext_abort_device_nGRE_prefetch_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_device_nGRE_read_ras_index

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`).

Type: `int`. Default value: `-0x1`.

ext_abort_device_nGRE_read_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_device_nGRE_write_ras_index

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`).

Type: `int`. Default value: `-0x1`.

ext_abort_device_nGRE_write_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_device_prefetch_ras_index

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`).

Type: `int`. Default value: `-0x1`.

ext_abort_device_prefetch_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_device_read_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number_of_error_records).

Type: `int`. Default value: `0x0`.

ext_abort_device_read_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_device_write_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number_of_error_records).

Type: `int`. Default value: `0x0`.

ext_abort_device_write_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_normal_cacheable_read_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number_of_error_records).

Type: `int`. Default value: `0x0`.

ext_abort_normal_cacheable_read_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_normal_cacheable_write_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number_of_error_records).

Type: `int`. Default value: `0x0`.

ext_abort_normal_cacheable_write_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_normal_noncacheable_prefetch_ras_index

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).

Type: `int`. Default value: `-0x1`.

ext_abort_normal_noncacheable_prefetch_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_normal_noncacheable_read_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number_of_error_records).

Type: `int`. Default value: `0x0`.

ext_abort_normal_noncacheable_read_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_normal_noncacheable_write_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number_of_error_records).

Type: `int`. Default value: `0x0`.

ext_abort_normal_noncacheable_write_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_normal_wt_cacheable_prefetch_ras_index

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).

Type: `int`. Default value: `-0x1`.

ext_abort_normal_wt_cacheable_prefetch_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_normal_wt_cacheable_read_ras_index

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).

Type: `int`. Default value: `-0x1`.

ext_abort_normal_wt_cacheable_read_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_normal_wt_cacheable_write_ras_index

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).

Type: `int`. Default value: `-0x1`.

ext_abort_normal_wt_cacheable_write_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_prefetch_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range `[0, number_of_error_records)`.

Type: `int`. Default value: `0x0`.

ext_abort_prefetch_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_so_prefetch_ras_index

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)`.

Type: `int`. Default value: `-0x1`.

ext_abort_so_prefetch_ras_type

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.

Type: `int`. Default value: `-0x1`.

ext_abort_so_read_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range `[0, number_of_error_records)`.

Type: `int`. Default value: `0x0`.

ext_abort_so_read_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_so_write_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range `[0, number_of_error_records)`.

Type: `int`. Default value: `0x0`.

ext_abort_so_write_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_ttw_cacheable_read_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range `[0, number_of_error_records)`.

Type: `int`. Default value: `0x0`.

ext_abort_ttw_cacheable_read_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_ttw_noncacheable_read_ras_index

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number_of_error_records).

Type: `int`. Default value: `0x0`.

ext_abort_ttw_noncacheable_read_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_ttw_wt_cacheable_read_ras_index

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).

Type: `int`. Default value: `-0x1`.

ext_abort_ttw_wt_cacheable_read_ras_type

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.

Type: `int`. Default value: `-0x1`.

fault_on_nT_bit_set

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.

Type: `bool`. Default value: `0x1`.

fault_unalign_to_unsupported_access

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.

Type: `int`. Default value: `0x8`.

force_deterministic_irg_tag_generation

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.

Type: `bool`. Default value: `0x0`.

force_mte_tag_access_razwi_and_ignore_tag_checks

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.

Type: `bool`. Default value: `0x0`.

force_zero_mpam_partid_and_pmg

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.

Type: `int`. Default value: `0x0`.

`fsr_ext_bit_update_kind`

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.

Type: `int`. Default value: `0x3`.

`gcs_data_check_overrides_data_abort`

GCS Data check exceptions are taken before Data Aborts.

Type: `bool`. Default value: `0x0`.

`GMID-log2-block-size`

Log2 of the block size accessed by STGM/LDGM/STZGM instructions.

Type: `int`. Default value: `0x4`.

`hardware_translation_table_update_implemented`

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.

Type: `int`. Default value: `0x1`.

`has_128_bit_atomic_instructions`

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.

Type: `int`. Default value: `0x0`.

`has_128_bit_tt_descriptors`

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.

Type: `int`. Default value: `0x0`.

`has_16bit_vmids`

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.

Type: `int`. Default value: `0x1`.

`has_52bit_address_with_16k`

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.

Type: `int`. Default value: `0x0`.

`has_52bit_address_with_4k`

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.

Type: `int`. Default value: `0x0`.

has_56_bit_va

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.

Type: `int`. Default value: `0x0`.

has_64bit_pmu_ext_access

Implement 64-bit pmu external interface access Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x0`.

has_aarch32_hpd

If true then hierarchical permission disable is supported in AArch32 (FEAT_AA32HPD).

Type: `bool`. Default value: `0x0`.

has_aarch64

All implemented exception levels can run in AArch64.

Type: `bool`. Default value: `0x1`.

has_actlr2

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR_EL1[63:32].

Type: `bool`. Default value: `0x0`.

has_actlr_virtualisation

If true ACTLR_EL12 is implemented and ACTLR_EL1 supports virtualisation.

Type: `bool`. Default value: `0x0`.

has_address_breakpoint_linking

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.

Type: `int`. Default value: `0x0`.

has_amu

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.

Type: `int`. Default value: `0x0`.

has_amu_ext64

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.

Type: `int`. Default value: `0x0`.

has_arm_v8-1

Implement the ARMv8.1 Extension.

Type: `bool`. Default value: `0x0`.

has_arm_v8-2

Implement the ARMv8.2 Extension.

Type: bool. Default value: 0x0.

has_arm_v8-3

Implement the ARMv8.3 Extension.

Type: bool. Default value: 0x0.

has_arm_v8-4

Implement the ARMv8.4 Extension.

Type: bool. Default value: 0x0.

has_arm_v8-5

Implement the ARMv8.5 Extension.

Type: bool. Default value: 0x0.

has_arm_v8-6

Implement the ARMv8.6 Extension.

Type: bool. Default value: 0x0.

has_arm_v8-7

Implement the Armv8.7 Extension.

Type: bool. Default value: 0x0.

has_arm_v8-8

Implement the ARMv8.8 Extension.

Type: bool. Default value: 0x0.

has_arm_v8-9

Implement the ARMv8.9 Extension. This feature is incomplete and under development.

Type: bool. Default value: 0x0.

has_asid2

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.

Type: int. Default value: 0x1.

has_at_with_pan

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.

Type: int. Default value: 0x1.

has_ats1a

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.

Type: int. Default value: 0x0.

has_attribute_index_enhancement

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.

Type: int. Default value: 0x0.

has_axflag_xaflag

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.

Type: `int`. Default value: `0x1`.

has_axflag_xaflag_frint

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.

Type: `int`. Default value: `0x1`.

has_branch_target_exception

Implement Branch target identification mechanism from ARMv8.5 (FEAT_BTI). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x0`.

has_ccidrx

Implement the ARMv8.3 CCSIDR Extension. Extending the `ccsidr` number of sets.

Type: `bool`. Default value: `0x0`.

has_cfinv_rmif_setf

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.

Type: `int`. Default value: `0x1`.

has_chkfeat

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.

Type: `int`. Default value: `0x0`.

has_clear_bhb

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.

Type: `int`. Default value: `0x1`.

has_clear_other_speculation_by_context

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.

Type: `int`. Default value: `0x1`.

has_cmo_wr_control

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.

Type: `int`. Default value: `0x1`.

has_common_not_private_translations

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.

Type: `int`. Default value: `0x1`.

has_complex_number

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.

Type: `int`. Default value: `0x1`.

has_const_pac

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.

Type: `int`. Default value: `0x0`.

has_cssc

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.

Type: `int`. Default value: `0x1`.

has_cvadp_support

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.

Type: `int`. Default value: `0x0`.

has_data_alignment_flag

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.

Type: `int`. Default value: `0x0`.

has_delayed_wfe_trap

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.

Type: `int`. Default value: `0x0`.

has_dgh

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.

Type: `int`. Default value: `0x1`.

has_dot_product

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.

Type: `int`. Default value: `0x1`.

has_e0pd

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.

Type: `int`. Default value: `0x1`.

has_e2h_rao

Whether the implementation treats HCR_EL2.E2H as Read-As-One (RAO). 0 : FEAT_E2HO implemented Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x0`.

has_ebep

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.

Type: `int`. Default value: `0x0`.

has_ebf16

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.

Type: `int`. Default value: `0x0`.

has_ecbhb

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.

Type: `int`. Default value: `0x1`.

has_edhshr

Implement external debug halt status register (FEAT_EDHSHR). 0: FEAT_EDHSHR is not implemented, 1: FEAT_EDHSHR is implemented, 2: FEAT_EDHSHR is implemented (extends EDHSHR to include the VNCR, CM, and WnR fields), 0xF: FEAT_EDHSHR implementation is dependent on FEAT_SME.

Type: `int`. Default value: `0x1`.

has_el2

Implements EL2.

Type: `bool`. Default value: `0x1`.

has_el3

Implements EL3.

Type: `bool`. Default value: `0x1`.

has_enhanced_pac

If pointer authentication is enabled then implement enhanced PAC (FEAT_EPAC).

Type: `bool`. Default value: `0x0`.

has_enhanced_pan

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.

Type: `int`. Default value: `0x1`.

has_enhanced_software_step

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.

Type: `int`. Default value: `0x1`.

has_ets

Whether Enhanced Translation Synchronization is supported (FEAT_ETS).

Type: `bool`. Default value: `0x0`.

has_extended_recv_estimate

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.

Type: `int`. Default value: `0x0`.

has_external_rndr

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.

Type: `int`. Default value: `0x0`.

has_fgt

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.

Type: `int`. Default value: `0x1`.

has_fgt2

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.

Type: `int`. Default value: `0x1`.

has_fgwt3

If true then Fine Grained Write EL3 is enabled (FEAT_FGWTE3).

Type: `bool`. Default value: `0x0`.

has_fixed_function_instr_counter

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.

Type: `int`. Default value: `0x0`.

has_fp16

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.

Type: `int`. Default value: `0x1`.

has_fp16_fmlal

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.

Type: `int`. Default value: `0x1`.

has_frint

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.

Type: `int`. Default value: `0x1`.

has_gcs

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.

Type: `int`. Default value: `0x0`.

has_generic_authentication

Implement ARMv8.3 generic authentication. Possible values of this parameter are: - 1, feature is implemented if ARMv8.3 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x1`.

has_guest_translation_granule

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.

Type: `int`. Default value: `0x1`.

has_haft

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.

Type: `int`. Default value: `0x0`.

has_hardware_accelerator_for_cleaning_dirty_state

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.

Type: `int`. Default value: `0x0`.

has_hardware_dirty_state_tracking_structure

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.

Type: `int`. Default value: `0x0`.

has_hardware_translation_table_update

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).

Type: `int`. Default value: `0x2`.

has_hcrx_el2

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.

Type: `int`. Default value: `0x1`.

has_hpmn0

Allow hypervisor to set MDCR_EL2.HPMN to 0 (FEAT_HPMN0) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x0`.

has_id_reg_read

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.

Type: `int`. Default value: `0x1`.

has_iesb

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.

Type: `int`. Default value: `0x1`.

has_ite

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.

Type: `int`. Default value: `0x0`.

has_jscvt

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.

Type: `int`. Default value: `0x1`.

has_large_ttbr_ba_without_lpa

When FEAT_LPA is not implemented, whether TTBR base address supports large values (52 bits) or not (48 bits).

Type: `bool`. Default value: `0x1`.

has_large_va

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.

Type: `int`. Default value: `0x0`.

has_ldapur_stlur

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.

Type: `int`. Default value: `0x1`.

has_ldm_stm_ordering_control

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.

Type: `int`. Default value: `0x0`.

has_lrcpc

If true then it support the RCpc feature from ARMv8.3 (FEAT_LRCPC).

Type: `bool`. Default value: `0x0`.

has_lrcpc3

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.

Type: `int`. Default value: `0x0`.

has_mismatch_and_range_breakpoints

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.

Type: `int`. Default value: `0x0`.

has_mismatch_watchpoints

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.

Type: `int`. Default value: `0x0`.

has_mops_option

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.

Type: `int`. Default value: `0x0`.

has_mpam

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.

Type: `int`. Default value: `0x0`.

has_mt_pmu_disable_feature

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_AA64DFRO_EL1.MTPMU.

Type: `int`. Default value: `0x0`.

has_mte_async_faults

Whether MTE asynchronous faults are supported (FEAT_MTE_ASYNC).

Type: `bool`. Default value: `0x1`.

has_mte_ctrl_bits_stateful

if `memory_tagging_support_level == 1`, Whether mte specific control bits in system registers are stateful.

Type: `bool`. Default value: `0x0`.

has_mte_perm

Implement tag access permission (FEAT_MTE_PERM).

Type: `bool`. Default value: `0x0`.

has_mte_tag_related_fault_high_prio_than_data

For DC GZVA, Whether MMU faults generated by tag access has higher priority than faults due to data access.

Type: `bool`. Default value: `0x0`.

has_nested_virtualization

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.

Type: `int`. Default value: `0x1`.

has_nmi

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.

Type: `int`. Default value: `0x1`.

has_no_os_double_lock

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.

Type: `int`. Default value: `0x0`.

has_non_context_synchronizing_exception_controls

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.

Type: `int`. Default value: `0x1`.

has_nv1_raz

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 ARMv9.5 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x0`.

has_nv_frac

Whether the NV_frac behavior is supported. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x0`.

has_par_bit10_razwi

Whether `PAR_EL1[10]` is RAZ/WI.

Type: `bool`. Default value: `0x0`.

has_pc_sample_profiling_enable

Whether PC Sample profiling enable is implemented (FEAT_PCSRv8p9).

Type: `bool`. Default value: `0x0`.

has_permission_indirection_s1

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.

Type: `int`. Default value: `0x0`.

has_permission_indirection_s2

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.

Type: `int`. Default value: `0x0`.

has_permission_overlay_s1

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.

Type: `int`. Default value: `0x0`.

has_permission_overlay_s2

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.

Type: `int`. Default value: `0x0`.

has_pmss

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.

Type: `int`. Default value: `0x0`.

has_pmu_edge_detection

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.

Type: `int`. Default value: `0x0`.

has_pmu_for_sme_extension

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.

Type: `int`. Default value: `0x0`.

has_pmu_threshold_linking_control

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.

Type: `int`. Default value: `0x0`.

has_pointer_authentication

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.

Type: `int`. Default value: `0x1`.

has_prediction_invalidation_instructions

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.

Type: `int`. Default value: `0x1`.

has_prfm_slc

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.

Type: `int`. Default value: `0x0`.

has_pstate_dit

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.

Type: `int`. Default value: `0x1`.

has_pstate_pan

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.

Type: `int`. Default value: `0x1`.

has_pstate_uao

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.

Type: `int`. Default value: `0x1`.

has_qarma3_pac

Supports QARMA3 pointer authentication algorithm (FEAT_PACQARMA3).

Type: `bool`. Default value: `0x0`.

has_ras_aderr

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.

Type: `int`. Default value: `0x0`.

has_ras_anerr

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.

Type: `int`. Default value: `0x0`.

has_ras_armv84_extension

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.

Type: `int`. Default value: `0x1`.

has_ras_armv89_double_fault

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.

Type: `int`. Default value: `0x0`.

has_ras_armv89_extension

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.

Type: `int`. Default value: `0x1`.

has_ras_critical_error

[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.

Type: `int`. Default value: `0x0`.

has_ras_delegated_serror_exceptions_for_el3

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.

Type: `int`. Default value: `0x1`.

has_ras_double_fault

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.

Type: `int`. Default value: `0x1`.

has_ras_fault_injection

[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.

Type: `int`. Default value: `0x0`.

has_ras_pfar

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.

Type: `int`. Default value: `0x0`.

has_ras_timestamp

[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.

Type: `int`. Default value: `0x0`.

has_restriction_on_speculative_data_loaded

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.

Type: `int`. Default value: `0x1`.

has_rme_gpc2

If true then RME GPC2 extension is enabled (FEAT_RME_GPC2).

Type: `bool`. Default value: `0x0`.

has_rndr

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.

Type: `int`. Default value: `0x0`.

has_rndr_trap

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.

Type: `int`. Default value: `0x0`.

has_rounding_doubling_multiply_add_subtract

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.

Type: `int`. Default value: `0x1`.

has_rprfm

Support for RPRFM hint instruction (FEAT_RPRFM).

Type: `bool`. Default value: `0x0`.

has_sctlr2

Implement SCTL2_ELx registers (FEAT_SCTL2). Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x1`.

has_sebep

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.

Type: `int`. Default value: `0x0`.

has_secure_el2

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.

Type: `int`. Default value: `0x1`.

has_self_hosted_trace_extension

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.

Type: `int`. Default value: `0x1`.

has_small_page_table

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.

Type: `int`. Default value: `0x1`.

has_spe_eft

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.

Type: `int`. Default value: `0x1`.

has_spe_fds

Implement SPE filter by data source from ARMv8.9 (FEAT_SPE_FDS). Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x1`.

has_spe_fpf

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.

Type: `int`. Default value: `0x1`.

Type: `int`. Default value: `0x1`.

has_speculation_barrier_inst

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.

Type: `int`. Default value: `0x1`.

has_stage2_fwb

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.

Type: `int`. Default value: `0x1`.

has_stage2_xnx

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.

Type: `int`. Default value: `0x1`.

has_stage2_xnx_in_aarch32

Implement the extended XN[1:0] stage 2 control from ARMv8.2 for Aarch32 (FEAT_XNX).

Type: `bool`. Default value: `0x1`.

has_statistical_profiling

Whether Statistical Based Profiling is implemented (FEAT_SPE).

Type: `bool`. Default value: `0x1`.

has_supersections

Whether VMSAv8-32 supersection to support more than 32-bit PA using short descriptor is implemented.

Type: `bool`. Default value: `0x1`.

has_sysinstr128

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.

Type: `int`. Default value: `0x0`.

has_sysreg128

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.

Type: `int`. Default value: `0x0`.

has_tcr2

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.

Type: `int`. Default value: `0x1`.

has_tidcp1

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.

Type: `int`. Default value: `0x1`.

has_tlb_pa_caching

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).

Type: `bool`. Default value: `0x0`.

has_tlbi_range

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.

Type: `int`. Default value: `0x1`.

has_tlbi_to_outer_shareable

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.

Type: `int`. Default value: `0x1`.

has_tlbi_ttl

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.

Type: `int`. Default value: `0x1`.

has_tlbiw

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.

Type: `int`. Default value: `0x0`.

has_translation_hardening

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.

Type: `int`. Default value: `0x0`.

has_trbe_ext

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.

Type: `int`. Default value: `0x0`.

has_unaligned_single_copy_atomicity

Implement support for SCTLR_ELx.nAA from ARMv8.4, and A64 atomic, exclusive and acquire/release instructions accessing unaligned bytes inside a 16byte window will not generate alignment fault (FEAT_LSE2). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x1`.

has_v8_4_debug_extension

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.

Type: `int`. Default value: `0x1`.

has_v8_4_pmu_extension

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.

Type: `int`. Default value: `0x1`.

has_v8_5_debug_over_power_down

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.

Type: `int`. Default value: `0x0`.

has_v8_5_pmu_extension

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.

Type: `int`. Default value: `0x1`.

has_v8_5_spe_extension

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.

Type: `int`. Default value: `0x1`.

has_v8_6_pmu_events

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.

Type: `int`. Default value: `0x1`.

has_v8_7_fp_enhancements

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.

Type: `int`. Default value: `0x1`.

has_v8_7_pmu_events

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.

Type: `int`. Default value: `0x1`.

has_v8_7_pmu_extension

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.

Type: `int`. Default value: `0x1`.

has_v8_7_spe_extension

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.

Type: `int`. Default value: `0x1`.

has_v8_7_spe_inverted_filtering

Where FEAT_SPEv1p2 is implemented, whether the inverting filtering by events.

Type: `bool`. Default value: `0x1`.

has_v8_7_spe_previous_branch_target

Where FEAT_SPEv1p2 is implemented, whether the optional branch target feature is implemented.

Type: `bool`. Default value: `0x1`.

has_v8_8_debug_extension

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.

Type: `int`. Default value: `0x1`.

has_v8_8_pmu_events

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.

Type: `int`. Default value: `0x1`.

has_v8_8_pmu_extension

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.

Type: `int`. Default value: `0x1`.

has_v8_8_spe_extension

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.

Type: `int`. Default value: `0x1`.

has_v8_9_debug_extension

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.

Type: `int`. Default value: `0x1`.

has_v8_9_pc_sample_based_profiling

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.

Type: `int`. Default value: `0x1`.

has_v8_9_pmu_events

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.

Type: `int`. Default value: `0x1`.

has_v8_9_pmu_extension

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.

Type: `int`. Default value: `0x1`.

has_v8_9_spe_extension

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.

Type: `int`. Default value: `0x1`.

has_v9_5_spe_extension

Implement support of SME to SPE from ARMv9.5 Possible values of this parameter are: - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x1`.

has_vnocr_el2

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.

Type: `int`. Default value: `0x1`.

has_wfet_and_wfit

Implements WFE and WFI with Timeout from Armv8.7 (FEAT_WFvT) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x1`.

has_xs

Implements Armv8.7 XS, TLBnXS, DSBnXS instruction (FEAT_XS). Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

Type: `int`. Default value: `0x1`.

hdbss_error_fault_type

Type of fault reported for HDBSS errors. 0 = precise exception, 1 = fault logged in HDBSSPROD_EL2.FSC (FEAT_HDBSS).

Type: `int`. Default value: `0x0`.

hvbar_reset_is_rvbar

If true then the reset value of HVBAR is RVBAR, if false the reset value is UNKNOWN.

Type: `bool`. Default value: `0x0`.

ignore_access_flag_update_by_at_ops

If true, AT operations do not update access flag.

Type: `bool`. Default value: `0x0`.

ignore_data_abt_on_af_update_by_at_ops

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.

Type: `bool`. Default value: `0x1`.

ignore_DBGPRCR_CWRR

Ignore writes to the deprecated DBGPRCR.CWRR bit.

Type: `bool`. Default value: `0x0`.

ignore_large_address_top_bits_in_page_walk

Whether page table bits [15:12] are ignored if `PA_SIZE < 52` and output address is configured `< 52` with large page.

Type: `bool`. Default value: `0x0`.

impdef_regs_and_unpred_from_implementation

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.

Type: `string`. Default value: `""`.

independent_cache_control_traps

Implement Independent Cache Control traps from ARMv8.5. 0, No support. 1, Supported but not for tlb maintenance instructions. 2, Full support. (FEAT_EVT).

Type: `int`. Default value: `0x0`.

insert_iesb_before_exception

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.

Type: `bool`. Default value: `0x0`.

is_mt_res0

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.

Type: `bool`. Default value: `0x0`.

ish_is_osh

Whether Innershareable is same as OuterShareable.

Type: `bool`. Default value: `0x0`.

max_32bit_el

Maximum exception level supporting AArch32 modes.

Type: `int`. Default value: `0x3`.

mdselr_le_16_bps_wps_behaviour

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.

Type: `int`. Default value: `0x0`.

memory.acp.AxCACHE_mask

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.

Type: `int`. Default value: `0x0`.

memory.acp.AxCACHE_pattern

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.

Type: `int`. Default value: `0x0`.

memory_tagging_support_level

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).

Type: `int`. Default value: `0x0`.

mops_cpy_block_size

Block size used for memcpy memory accesses.

Type: `int`. Default value: `0x40`.

mops_cpy_default_dir

Default direction for non-overlapping memcpy operations: 0, forwards. 1, backwards.

Type: `int`. Default value: `0x0`.

mops_cpy_handle_async_exceptions

Handle any pending async exceptions after copying a block of data, instead of waiting until instruction end.

Type: `bool`. Default value: `0x0`.

mops_cpy_post_size

Percentage of data copied in memcpy 'E' instructions.

Type: `int`. Default value: `0xa`.

mops_cpy_pre_size

Percentage of data copied in memcpy 'P' instructions.

Type: `int`. Default value: `0xa`.

mops_cpy_pre_size_threshold

Size threshold in Bytes for CPY*P* instructions.

Type: `int`. Default value: `0x0`.

mops_cpy_single_access

Execute memcpy as a single read and single write access.

Type: `bool`. Default value: `0x0`.

mops_cpy_write_abort_before_read

Report the data aborts and watchpoint of the write accesses, before those of the read accesses.

Type: `bool`. Default value: `0x0`.

mops_cpy_zero_size_can_fault

Fault because of mismatched implementation option when the operation is of size 0.

Type: `bool`. Default value: `0x1`.

mops_exec_order_can_fault

Enable exception on the Main/Epilogue instruction when executed after a mismatched Prologue/Main in a CPY/SET sequence, or after another random instruction.

Type: `bool`. Default value: `0x0`.

mops_inst_cpy_zero_size_can_fault

Fault because of mismatched implementation option when `inst_cpy_size` is 0.

Type: `bool`. Default value: `0x1`.

mops_inst_set_zero_size_can_fault

Fault because of mismatched implementation option when `inst_set_size` is 0.

Type: `bool`. Default value: `0x1`.

mops_mmu_abort_far_aligned

If true, in case of an MMU abort on a MOPS instruction, report FAR aligned to current translation granule.

Type: `bool`. Default value: `0x0`.

mops_set_block_size

Block size used for memset memory accesses.

Type: `int`. Default value: `0x40`.

mops_set_handle_async_exceptions

Handle any pending async exceptions after setting a block of data, instead of waiting until instruction end.

Type: `bool`. Default value: `0x0`.

mops_set_post_size

Percentage of data copied in memset 'E' instructions.

Type: `int`. Default value: `0xa`.

mops_set_pre_size

Percentage of data copied in memset 'P' instructions.

Type: `int`. Default value: `0xa`.

mops_set_single_access

Execute memset as a single read and single write access.

Type: `bool`. Default value: `0x0`.

mops_set_zero_size_can_fault

Fault because of mismatched implementation option when the operation is of size 0.

Type: `bool`. Default value: `0x1`.

mops_setg_unaligned_does_mismatch_fault

If true, in case of unaligned SETGM / SETGE, raise a mismatched memset exception because of impdef reasons, instead of alignment fault.

Type: `bool`. Default value: `0x0`.

mops_wp_far_behaviour

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.

Type: `int`. Default value: `0x0`.

mpam_force_ns_rao

Whether MPAM3_EL3.FORCE_NS bit is RAO/WI.

Type: `bool`. Default value: `0x0`.

mpam_frac

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.

Type: `int`. Default value: `0x0`.

mpam_has_altsp

MPAM Whether MPAMIDR_EL1.HAS_ALTSP bit is set or clear.

Type: `bool`. Default value: `0x0`.

mpam_has_hcr

MPAM Whether MPAMIDR_EL1 HAS_HCR bit is set or clear.

Type: `bool`. Default value: `0x0`.

mpam_max_partid

MPAM Maximum PARTID Supported.

Type: `int`. Default value: `0xffff`.

mpam_max_pmg

MPAM Maximum PMG Supported.

Type: `int`. Default value: `0xff`.

mpam_max_vpmr

MPAM Maximum VPMR Supported.

Type: `int`. Default value: `0x0`.

mpamidr_has_force_ns

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.

Type: `int`. Default value: `0x0`.

mpamidr_has_sdeflt

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.

Type: `int`. Default value: `0x0`.

mpamidr_has_tidr

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.

Type: `int`. Default value: `0x0`.

mte_report_which_failed_address

Set to "first" or "last". Applicable only for MTE synchronous check. If "first" then report first failed address. If "last" then report the first address of last failed MTE granule or first address of transfer if it doesn't cross a granule boundary.

Type: `string`. Default value: "first".

mte_tminline

Value of CTR_ELO.TminLine for reading purpose only. A value configured using this does not indicate the presence of separate tag cache. 0, TminLine evaluated from smallest data cache line.

Type: `int`. Default value: `0x0`.

mvbar_reset_is_rvbar

If true then the reset value of MVBAR is RVBAR, if false the reset value is UNKNOWN.

Type: `bool`. Default value: `0x1`.

NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

num_loregion_descriptors

Number of Limited Ordering Region descriptors implemented (if ARM v8.1 extensions are implemented) (FEAT_LOR).

Type: `int`. Default value: `0x0`.

num_loregions

Number of Limited Ordering Regions implemented excluding background region (if ARM v8.1 extensions are implemented) (FEAT_LOR).

Type: `int`. Default value: `0x0`.

number_of_abl_breakpoints

if FEAT_ABLE is implemented, Number of address matching breakpoints that support address linking.

Type: `int`. Default value: `0x0`.

number_of_error_records

Cores Number of Error records supported for RAS.

Type: `int`. Default value: `0x0`.

optimal_alignment_size

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] .

Type: `int`. Default value: `0x1`.

output_attributes

User-defined transform to be applied to bus attributes like MasterID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`. Default value: "ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS".

PA_SIZE

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

Type: `int`. Default value: `0x28`.

pacm_support_level

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.

Type: `int`. Default value: `0x2`.

page_based_hardware_attributes

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).

Type: `int`. Default value: `0x0`.

pfar_is_valid

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.

Type: `bool`. Default value: `0x1`.

pfr1_csv2_frac

Fractional revision number ID_AA64PFR1_EL1.CSV2_frac when ID_AA64PFR0_EL1.CSV==1 for CSV2 extension (FEAT_CSV2_1p1, FEAT_CSV2_1p2).

Type: `int`. Default value: `0x0`.

pmmir_ell_bus_slots

Largest value by which BUS_ACCESS can increment over BUS_CYCLES cycles. From v8.7 PMU extension.

Type: `int`. Default value: `0x0`.

pmmir_ell_bus_width

Width, in bytes, of accesses counted by BUS_ACCESS. From v8.7 PMU extension.

Type: `int`. Default value: `0x0`.

pmu_async_exception_delay

Configure PMU asynchronous exception delay in CPU cycles (FEAT_SEBEP).

Type: `int`. Default value: `0x0`.

pmu_precise_events

"Configure v9.4 Precise PMU events. {"pmu_events":

["SW_INCR","PC_WRITE_RETIRED","BR_RETIRED","BR_IND_RETIRED","BR_RETURN_RETIRED","BR_RETU

Type: `string`. Default value: `""`.

pmu_threshold_bit_width

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.

Type: `int`. Default value: `0x0`.

pseudo_fault_generation_feature_register

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,

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.

Type: `string`. Default value: `""`.

pstate_ssbs_reset

Reset value of pstate.ssbs.

Type: `bool`. Default value: `0x0`.

pstate_ssbs_type

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).

Type: `int`. Default value: `0x0`.

ras_aderr_anerr_controls_are_same

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).

Type: `bool`. Default value: `0x0`.

ras_err_registers_undef_if_no_error_records

If true, all RAS error record registers, along with ERRSELR_EL1, will be undefined if ERRIDR_EL1 indicates that zero error records are implemented.

Type: `bool`. Default value: `0x0`.

ras_errselr_undef_if_no_error_records

If true, ERRSELR_EL1 will be undefined if ERRIDR_EL1 indicates that zero error records are implemented.

Type: `bool`. Default value: `0x0`.

ras_extra_configurations

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}].

Type: `string`. Default value: `""`.

ras_frac

0, No additional feature implemented. 1, Additional ERXMISC*, ERXPFG* registers and FaultInjection trap from RAS v1.1. implemented.

Type: `int`. Default value: `0x0`.

ras_log2_fault_granule_size

Log2 of the RAS fault granule size in KB.

Type: `int`. Default value: `0x2`.

ras_pfg_clock_mhz

RAS Pseudo-Fault generation clock rate in MHz.

Type: `int`. Default value: `0x18`.

ras_report_aligned_pa_in_pfar

If true, the PFAR_ELx register reports the PA aligned to the RAS fault granule size on a sync external abort or SError exception.

Type: bool. Default value: 0x0.

restriction_on_speculative_execution

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.

Type: int. Default value: 0x0.

restriction_on_speculative_execution_aarch32

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).

Type: int. Default value: 0x0.

rnr_always_implemented

Always implement RNR_ELx, RNR, or HRNR at the highest implemented exception level, even if that exception level cannot use both AArch32 and AArch64.

Type: bool. Default value: 0x0.

rndr_rndrrs_seed

Initial seed for random engine used in RNR register.

Type: int. Default value: 0x0.

set_mops_option

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.

Type: int. Default value: 0x0.

set_rasv10_for_armv84_and_higher

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.

Type: bool. Default value: 0x0.

setg_mops_option

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.

Type: `int`. Default value: `0x0`.

spe_counter_size

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.

Type: `int`. Default value: `0x1`.

spmu_support_level

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.

Type: `int`. Default value: `0x0`.

spsr_el3_is_mapped_to_spsr_mon

Whether SPSR_EL3 is mapped to AArch32 register SPSR_mon.

Type: `bool`. Default value: `0x1`.

spsr_m4_res0

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

Type: `bool`. Default value: `0x0`.

stage12_tlb_size

Number of stage1+2 tlb entries.

Type: `int`. Default value: `0x80`.

stage2_tlb_size

Number of stage2 only tlb entries.

Type: `int`. Default value: `0x0`.

stage2_walkcache_size

Number of stage2 only walk cache entries.

Type: `int`. Default value: `0x0`.

statistical_profiling_buffer_alignment

Statistical profiling alignment constraint for sample buffer.

Type: `int`. Default value: `0x1`.

statistical_profiling_random_interval_is_separate

Statistical profiling random interval gets added to the main timer interval(false) or (true) runs as separate timer.

Type: `bool`. Default value: `0x0`.

statistical_profiling_recommended_min_sampling

Statistical profiling recommended minimum sampling interval.

Type: `int`. Default value: `0x100`.

store_excl_fail_tag_check_action

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.

Type: `int`. Default value: `0x0`.

stzgm_reports_fault_address_from_reg_arg

Whether STZGM reports register arg as fault address in FAR.

Type: `bool`. Default value: `0x0`.

sync_ext_abort_is_sync_serror

Treat synchronous external aborts as synchronous SErrors (RASv8.9). 0, synchronous external abort. 1, synchronous serror.

Type: `bool`. Default value: `0x0`.

system_pmu_id

When FEAT_SPMU is implemented, indicates the largest value `s` to select a System PMU `<s>`.

Type: `int`. Default value: `0x0`.

tcr_ps_reserved_value_size

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.

Type: `int`. Default value: `0x0`.

tcr_txsz_undersize_should_fault

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`. Default value: `0x1`.

tdosa_traps_osdlr_if_no_os_double_lock

MDCR_EL*.TDOSA enables trap on OSDLR_EL1 and DBGOSDLR when OS double-lock is not implemented.

Type: `bool`. Default value: `0x1`.

trace_full_simd_reg_with_nep

Whether full simd register is traced even if partial update is done when FPCR.NEP=1.

Type: `bool`. Default value: `0x0`.

trace_icc_registers_as_icv_when_redirected

If true, update trace with ICV, instead of ICC when ICV registers are accessed depending on the core state.

Type: `bool`. Default value: `0x0`.

trace_physical_registers_when_host_virtualisation_enabled

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.

Type: `int`. Default value: `0x0`.

treat_PAC_as_NOP

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.

Type: `bool`. Default value: `0x0`.

undef_ccsidr2_access_for_unimplemented_aarch32

Whether access to CCSIDR2 is undef if AArch32 is implemented or not at EL1.

Type: bool. Default value: 0x0.

unpred_mte_tag_read_when_ata_controls_are_zero_or_untagged_attr

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.

Type: bool. Default value: 0x0.

unpred_mte_tag_store_data_cache_instr_to_device_mem_as_alignment_fault

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.

Type: bool. Default value: 0x0.

unpred_mte_tag_store_to_device_mem_as_alignment_fault

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.

Type: bool. Default value: 0x0.

unpred_nested_virtualization_nv_behaviour

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 .

Type: int. Default value: 0x0.

unpred_vnchr_el2_ress_mismatch

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 .

Type: int. Default value: 0x0.

unpred_zero_spsr_btype

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.

Type: bool. Default value: 0x1.

use_mte_eac_02_instructions_encoding

Use new MTE Instructions encoding since MTE spec EAC 0.2 in ARMv8.5.

Type: bool. Default value: 0x1.

use_mte_eac_08_tfsr_encoding

Use new MTE TFSR_ELx encodings since MTE spec EAC 0.8 in ARMv8.5.

Type: `bool`. Default value: `0x1`.

use_rosetta_disass

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.

Type: `int`. Default value: `0x1`.

use_sif_to_compute_pan

Where FEAT_PAN3 is implemented, whether SCR_EL3.SIF bit is used to determine instruction access permission for the purpose of PAN.

Type: `bool`. Default value: `0x0`.

use_Xt_as_LDG_STG_input

Use new MTE Instructions formats for LDG/ST(Z)(2)G, which use Xt as an input.

Type: `bool`. Default value: `0x1`.

vpu_datapath_width

VPU data path width.

Type: `int`. Default value: `0x80`.

wp_num_reporting

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.

Type: `int`. Default value: `0x0`.

Related information

[ScalableVectorExtension](#) on page 1658

4.5.3 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 4-78: IP revisions support

Revision	Quality level
v8.0M	Full support
v8.1M	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMAEMv8MCT contains the following CADI targets:

- ARM_AEMv8M

ARMAEMv8MCT contains the following MTI components:

- [ARM_AEMv8M](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)

Ports for ARMAEMv8MCT

Table 4-79: 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

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.

Parameters for ARM_AEMv8M

AIRCR.ENDIANNESS

Initialize processor to big endian mode.

Type: bool. Default value: 0x0.

allow_debug_monitor_with_in_flight_inst

Allow handling Debug Monitor exception with in-flight instructions.

Type: bool. Default value: 0x0.

allow_stack_accesses_to_ppb_space

Allow stack accesses to PPB space.

Type: bool. Default value: 0x0.

baseline

Use the baseline profile (if false, use mainline).

Type: bool. Default value: 0x1.

BB_PRESENT

Enable bitbanding.

Type: bool. Default value: 0x0.

callee_register_push_low_to_high

If true, push callee registers in order from R4 to R11. If false, push R11 to R4.

Type: bool. Default value: 0x1.

cde_fp_check_on_unsupported

Run FP checks on both supported and unsupported CDE instructions.

Type: bool. Default value: 0x0.

cde_impl_name

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

Type: string. Default value: "".

CFGMEMALIAS

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.

Type: int. Default value: 0x0.

CFGPAHBSZ

Size of the P-AHB peripheral port memory region. 0=P-AHB disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

Type: int. Default value: 0x0.

cpi_div

divider for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

cpi_mul

multiplier for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

CTI

CTI (Cross Trigger Interface) included.
Type: `bool`. Default value: `0x0`.

CTI_irq0_pin

CTI interrupt request 0 pin.
Type: `int`. Default value: `0x4`.

CTI_irq1_pin

CTI interrupt request 1 pin.
Type: `int`. Default value: `0x5`.

dcache-invalidate-ns-cleans-s

Whether V8M DCI* in non-secure should clean-and-invalidate secure cache contents.
Type: `bool`. Default value: `0x0`.

dcache-size

L1 D-cache size in bytes.
Type: `int`. Default value: `0x8000`.

dcache-state_modelled

Set whether D-cache has stateful implementation.
Type: `bool`. Default value: `0x0`.

dcache-ways

L1 D-cache ways (sets are implicit from size).
Type: `int`. Default value: `0x4`.

delay_faultmask_update

Delay FAULTMASK update to context sync.
Type: `bool`. Default value: `0x0`.

delay_sysreg_update

Delay some system register updates (e.g. SHCSR) to context sync.
Type: `bool`. Default value: `0x0`.

dtcm_enable

Enable DTCM at reset.
Type: `bool`. Default value: `0x0`.

dtcm_size

DTCM size in KB.
Type: `int`. Default value: `0x100`.

DTGU

DTCM Security Gate Unit included.

Type: `bool`. Default value: `0x0`.

DTGUBLKSZ

DTCM gate unit block size. Size= $\text{pow}(2, \text{DTGUBLKSZ} + 5)$ bytes.

Type: `int`. Default value: `0x3`.

DTGUMAXBLKS

Maximum number of DTCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{DTGUMAXBLKS})$.

Type: `int`. Default value: `0x0`.

duplicate_CADI_TCM_writes

CADI writes to TCMs are also sent to downstream memory at same addresses (for validation platforms).

Type: `bool`. Default value: `0x0`.

DWT_CTRL.NOCYCCNT

DWT cycle-counter not present (v8M-bl/v6M never have one).

Type: `bool`. Default value: `0x0`.

DWT_CTRL.NOPRFCNT

DWT performance-counters not present (v8M-bl/v6M never have them).

Type: `bool`. Default value: `0x0`.

DWT_CTRL.NUMCOMP

Number of watchpoint unit comparators implemented.

Type: `int`. Default value: `0x4`.

DWT_DEVARCH.REVISION

0: V2, 1: V2.1.

Type: `int`. Default value: `0x1`.

DWT_FUNCTION0.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION0. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0xb`.

DWT_FUNCTION1.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION1. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0x1e`.

DWT_FUNCTION10.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION10. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0xb`.

DWT_FUNCTION11.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION11. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0x1e`.

DWT_FUNCTION12.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION12. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0xb`.

DWT_FUNCTION13.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION13. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0x1e`.

DWT_FUNCTION14.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION14. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0xb`.

DWT_FUNCTION15.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION15. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0x1e`.

DWT_FUNCTION2.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION2. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0xb`.

DWT_FUNCTION3.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION3. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0x1e`.

DWT_FUNCTION4.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION4. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0xb`.

DWT_FUNCTION5.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION5. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0x1e`.

DWT_FUNCTION6.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION6. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0xb`.

DWT_FUNCTION7.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION7. If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0x1e`.

DWT_FUNCTION8.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION8.
If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0xb`.

DWT_FUNCTION9.ID

Sets the capabilities of the comparator that is accessible via the register, DWT_FUNCTION9.
If 'baseline' is set, invalid ID bits are cleared.

Type: `int`. Default value: `0x1e`.

DWT_TRACE

Support for DWT trace, controls the DWT_CTRL.NOTRCPKT bit. `false` : No DWT trace included, `true`: DWT trace included.

Type: `bool`. Default value: `0x1`.

dwt_unaligned_word_access_as_half_word

DWT Treat unaligned word access as half word or bytes.

Type: `bool`. Default value: `0x1`.

exercise_strex_fail

Reject a pseudo-random majority of exclusive store instructions.

Type: `bool`. Default value: `0x0`.

FP_CTRL.NUM_CODE

Number of breakpoint unit comparators implemented (limited to 15 in V6M or baseline).

Type: `int`. Default value: `0x8`.

FP_CTRL.NUM_LIT

How many Literals FPB supports remapping (ignored if baseline or TZM).

Type: `int`. Default value: `0x0`.

FP_REMAP.RMPSPPT

FPB supports remapping (ignored if baseline or SECEXT).

Type: `bool`. Default value: `0x1`.

FPB_HAS_LSR

FPB has LAR and LSR for software lock if mainline.

Type: `bool`. Default value: `0x1`.

has_arm_v8-1m

Enable v8.1M architecture version and features.

Type: `bool`. Default value: `0x0`.

has_cde

Enables Custom Datapath Extensions.

Type: `bool`. Default value: `0x0`.

has_writebuffer

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

Type: `bool`. Default value: `0x0`.

icache-size

L1 I-cache size in bytes.

Type: `int`. Default value: `0x8000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

icache-ways

L1 I-cache ways (sets are implicit from size).

Type: `int`. Default value: `0x2`.

ID_DFR0.Debug_Model_M_profile

Set whether debug extensions are implemented.

Type: `bool`. Default value: `0x1`.

ID_ISAR5.PACBTI

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.

Type: `int`. Default value: `0x0`.

ignore_demcr_sdme_for_nonhalting_bkpt

Ignore the SDME bit of the DEMCR register when escalating a Debug Monitor exception to a HardFault.

Type: `bool`. Default value: `0x0`.

ignore_out_of_range_RNR_write

If an MPU_RNR.REGION write is out of range, ignore it ; if false, MPU_RNR values wrap.

Type: `bool`. Default value: `0x0`.

ignore_RNR_top_nibble

If set, only the bottom four bits of MPU_RNR.REGION are used.

Type: `bool`. Default value: `0x0`.

INITPAHBEN

The P-AHB enable state at reset.

Type: `bool`. Default value: `0x0`.

INITVTOR_NS

Non-Secure vector-table offset at reset.

Type: `int`. Default value: `0x0`.

INITVTOR_S

Secure vector-table offset at reset.

Type: `int`. Default value: `0x0`.

iss_env

Enable ISSComp logic.

Type: `bool`. Default value: `0x0`.

itcm_enable

Enable ITCM at reset.

Type: `bool`. Default value: `0x0`.

itcm_size

ITCM size in KB.

Type: `int`. Default value: `0x100`.**ITGU**

ITCM Security Gate Unit included.

Type: `bool`. Default value: `0x0`.**ITGUBLKSZ**ITCM gate unit block size. Size= $\text{pow}(2, \text{ITGUBLKSZ} + 5)$ bytes.Type: `int`. Default value: `0x3`.**ITGUMAXBLKS**Maximum number of ITCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{ITGUMAXBLKS})$.Type: `int`. Default value: `0x0`.**ITM**Level of instrumentation trace supported. `false` : No ITM trace included, `true`: ITM trace included (unless baseline).Type: `bool`. Default value: `0x1`.**ITM_HAS_LSR**

ITM support LAR and LSR for software lock.

Type: `bool`. Default value: `0x1`.**LOCKDTGU**

Lock down of Data TGU registers write.

Type: `bool`. Default value: `0x0`.**LOCKITGU**

Lock down of Instruction TGU registers write.

Type: `bool`. Default value: `0x0`.**LOCKTCM**

Lock down of TCM registers write.

Type: `bool`. Default value: `0x0`.**LVL_WIDTH**

Number of bits of interrupt priority (baseline has 2).

Type: `int`. Default value: `0x3`.**master_id**

Master ID presented in bus transactions.

Type: `int`. Default value: `0x0`.**min_sync_level**

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

Type: `int`. Default value: `0x0`.**MPU_TYPE_NS.DREGION**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

Type: `int`. Default value: `0x10`.

MPU_TYPE_S.DREGION

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

Type: `int`. Default value: `0x10`.

MVE

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.

Type: `int`. Default value: `0x2`.

mve_has_atomic_ticks

Enable atomic ticks behaviour for vector instructions flagged as such (e.g. VLDR).

Type: `bool`. Default value: `0x0`.

MVFR0.Double-precision

Support 8-byte floats.

Type: `bool`. Default value: `0x1`.

NUM_IRQ

Number of user interrupts.

Type: `int`. Default value: `0x10`.

number_of_itm_stimulus_ports

The number of ITM stimulus ports.

Type: `int`. Default value: `0x20`.

rd_ns_bus_err_behave

External read aborts in nonsecure domain 0:ignored, 1:precise, 2:imprecise, 3=imprecise except SO.

Type: `int`. Default value: `0x1`.

rd_s_bus_err_behave

External read aborts in secure domain 0:ignored, 1:precise, 2:imprecise, 3=imprecise except SO.

Type: `int`. Default value: `0x1`.

REGISTER_POP_ORDER

Order in which the registers are popped off the stack during exception return. A comma separated list of register names and ranges.

Type: `string`. Default value: `"R4-R11,R0-R3,R12,R14,RETURN_ADDR,CPSR,S0-S15,FPSCR,PADDING,S16-S31"`.

REGISTER_PUSH_ORDER

Order in which the registers are pushed on to the stack during exception handling. A comma separated list of register names and ranges.

Type: `string`. Default value: `"R0-R3,R12,R14,RETURN_ADDR,CPSR,S0-S15,FPSCR,PADDING,S16-S31"`.

SAU_CTRL.ALLNS

At reset, the SAU treats entire memory space as NS when the SAU is disabled if this is set.

Type: `bool`. Default value: `0x0`.

SAU_CTRL.ENABLE

Enable SAU at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION0.BADDR

Base address of SAU region0 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION0.ENABLE

Enable SAU region0 at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION0.LADDR

Limit address of SAU region0 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION0.NSC

Set NSC for SAU region0 at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION1.BADDR

Base address of SAU region1 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION1.ENABLE

Enable SAU region1 at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION1.LADDR

Limit address of SAU region1 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION1.NSC

Set NSC for SAU region1 at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION10.BADDR

Base address of SAU region10 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION10.ENABLE

Enable SAU region10 at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION10.LADDR

Limit address of SAU region10 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION10.NSC

Set NSC for SAU region10 at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION11.BADDR

Base address of SAU region11 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION11.ENABLE

Enable SAU region11 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION11.LADDR

Limit address of SAU region11 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION11.NSC

Set NSC for SAU region11 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION12.BADDR

Base address of SAU region12 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION12.ENABLE

Enable SAU region12 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION12.LADDR

Limit address of SAU region12 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION12.NSC

Set NSC for SAU region12 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION13.BADDR

Base address of SAU region13 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION13.ENABLE

Enable SAU region13 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION13.LADDR

Limit address of SAU region13 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION13.NSC

Set NSC for SAU region13 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION14.BADDR

Base address of SAU region14 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION14.ENABLE

Enable SAU region14 at reset.

Type: bool. Default value: 0x0.

SAU_REGION14.LADDR

Limit address of SAU region14 at reset.

Type: int. Default value: 0x0.

SAU_REGION14.NSC

Set NSC for SAU region14 at reset.

Type: bool. Default value: 0x0.

SAU_REGION15.BADDR

Base address of SAU region15 at reset.

Type: int. Default value: 0x0.

SAU_REGION15.ENABLE

Enable SAU region15 at reset.

Type: bool. Default value: 0x0.

SAU_REGION15.LADDR

Limit address of SAU region15 at reset.

Type: int. Default value: 0x0.

SAU_REGION15.NSC

Set NSC for SAU region15 at reset.

Type: bool. Default value: 0x0.

SAU_REGION16.BADDR

Base address of SAU region16 at reset.

Type: int. Default value: 0x0.

SAU_REGION16.ENABLE

Enable SAU region16 at reset.

Type: bool. Default value: 0x0.

SAU_REGION16.LADDR

Limit address of SAU region16 at reset.

Type: int. Default value: 0x0.

SAU_REGION16.NSC

Set NSC for SAU region16 at reset.

Type: bool. Default value: 0x0.

SAU_REGION17.BADDR

Base address of SAU region17 at reset.

Type: int. Default value: 0x0.

SAU_REGION17.ENABLE

Enable SAU region17 at reset.

Type: bool. Default value: 0x0.

SAU_REGION17.LADDR

Limit address of SAU region17 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION17.NSC

Set NSC for SAU region17 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION18.BADDR

Base address of SAU region18 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION18.ENABLE

Enable SAU region18 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION18.LADDR

Limit address of SAU region18 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION18.NSC

Set NSC for SAU region18 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION19.BADDR

Base address of SAU region19 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION19.ENABLE

Enable SAU region19 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION19.LADDR

Limit address of SAU region19 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION19.NSC

Set NSC for SAU region19 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION2.BADDR

Base address of SAU region2 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION2.ENABLE

Enable SAU region2 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION2.LADDR

Limit address of SAU region2 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION2.NSC

Set NSC for SAU region2 at reset.
Type: bool. Default value: 0x0.

SAU_REGION20.BADDR

Base address of SAU region20 at reset.
Type: int. Default value: 0x0.

SAU_REGION20.ENABLE

Enable SAU region20 at reset.
Type: bool. Default value: 0x0.

SAU_REGION20.LADDR

Limit address of SAU region20 at reset.
Type: int. Default value: 0x0.

SAU_REGION20.NSC

Set NSC for SAU region20 at reset.
Type: bool. Default value: 0x0.

SAU_REGION21.BADDR

Base address of SAU region21 at reset.
Type: int. Default value: 0x0.

SAU_REGION21.ENABLE

Enable SAU region21 at reset.
Type: bool. Default value: 0x0.

SAU_REGION21.LADDR

Limit address of SAU region21 at reset.
Type: int. Default value: 0x0.

SAU_REGION21.NSC

Set NSC for SAU region21 at reset.
Type: bool. Default value: 0x0.

SAU_REGION22.BADDR

Base address of SAU region22 at reset.
Type: int. Default value: 0x0.

SAU_REGION22.ENABLE

Enable SAU region22 at reset.
Type: bool. Default value: 0x0.

SAU_REGION22.LADDR

Limit address of SAU region22 at reset.
Type: int. Default value: 0x0.

SAU_REGION22.NSC

Set NSC for SAU region22 at reset.
Type: bool. Default value: 0x0.

SAU_REGION23.BADDR

Base address of SAU region23 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION23.ENABLE

Enable SAU region23 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION23.LADDR

Limit address of SAU region23 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION23.NSC

Set NSC for SAU region23 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION24.BADDR

Base address of SAU region24 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION24.ENABLE

Enable SAU region24 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION24.LADDR

Limit address of SAU region24 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION24.NSC

Set NSC for SAU region24 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION25.BADDR

Base address of SAU region25 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION25.ENABLE

Enable SAU region25 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION25.LADDR

Limit address of SAU region25 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION25.NSC

Set NSC for SAU region25 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION26.BADDR

Base address of SAU region26 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION26.ENABLE

Enable SAU region26 at reset.

Type: bool. Default value: 0x0.

SAU_REGION26.LADDR

Limit address of SAU region26 at reset.

Type: int. Default value: 0x0.

SAU_REGION26.NSC

Set NSC for SAU region26 at reset.

Type: bool. Default value: 0x0.

SAU_REGION27.BADDR

Base address of SAU region27 at reset.

Type: int. Default value: 0x0.

SAU_REGION27.ENABLE

Enable SAU region27 at reset.

Type: bool. Default value: 0x0.

SAU_REGION27.LADDR

Limit address of SAU region27 at reset.

Type: int. Default value: 0x0.

SAU_REGION27.NSC

Set NSC for SAU region27 at reset.

Type: bool. Default value: 0x0.

SAU_REGION28.BADDR

Base address of SAU region28 at reset.

Type: int. Default value: 0x0.

SAU_REGION28.ENABLE

Enable SAU region28 at reset.

Type: bool. Default value: 0x0.

SAU_REGION28.LADDR

Limit address of SAU region28 at reset.

Type: int. Default value: 0x0.

SAU_REGION28.NSC

Set NSC for SAU region28 at reset.

Type: bool. Default value: 0x0.

SAU_REGION29.BADDR

Base address of SAU region29 at reset.

Type: int. Default value: 0x0.

SAU_REGION29.ENABLE

Enable SAU region29 at reset.

Type: bool. Default value: 0x0.

SAU_REGION29.LADDR

Limit address of SAU region29 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION29.NSC

Set NSC for SAU region29 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION3.BADDR

Base address of SAU region3 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION3.ENABLE

Enable SAU region3 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION3.LADDR

Limit address of SAU region3 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION3.NSC

Set NSC for SAU region3 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION30.BADDR

Base address of SAU region30 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION30.ENABLE

Enable SAU region30 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION30.LADDR

Limit address of SAU region30 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION30.NSC

Set NSC for SAU region30 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION31.BADDR

Base address of SAU region31 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION31.ENABLE

Enable SAU region31 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION31.LADDR

Limit address of SAU region31 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION31.NSC

Set NSC for SAU region31 at reset.
Type: bool. Default value: 0x0.

SAU_REGION32.BADDR

Base address of SAU region32 at reset.
Type: int. Default value: 0x0.

SAU_REGION32.ENABLE

Enable SAU region32 at reset.
Type: bool. Default value: 0x0.

SAU_REGION32.LADDR

Limit address of SAU region32 at reset.
Type: int. Default value: 0x0.

SAU_REGION32.NSC

Set NSC for SAU region32 at reset.
Type: bool. Default value: 0x0.

SAU_REGION33.BADDR

Base address of SAU region33 at reset.
Type: int. Default value: 0x0.

SAU_REGION33.ENABLE

Enable SAU region33 at reset.
Type: bool. Default value: 0x0.

SAU_REGION33.LADDR

Limit address of SAU region33 at reset.
Type: int. Default value: 0x0.

SAU_REGION33.NSC

Set NSC for SAU region33 at reset.
Type: bool. Default value: 0x0.

SAU_REGION34.BADDR

Base address of SAU region34 at reset.
Type: int. Default value: 0x0.

SAU_REGION34.ENABLE

Enable SAU region34 at reset.
Type: bool. Default value: 0x0.

SAU_REGION34.LADDR

Limit address of SAU region34 at reset.
Type: int. Default value: 0x0.

SAU_REGION34.NSC

Set NSC for SAU region34 at reset.
Type: bool. Default value: 0x0.

SAU_REGION35.BADDR

Base address of SAU region35 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION35.ENABLE

Enable SAU region35 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION35.LADDR

Limit address of SAU region35 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION35.NSC

Set NSC for SAU region35 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION36.BADDR

Base address of SAU region36 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION36.ENABLE

Enable SAU region36 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION36.LADDR

Limit address of SAU region36 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION36.NSC

Set NSC for SAU region36 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION37.BADDR

Base address of SAU region37 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION37.ENABLE

Enable SAU region37 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION37.LADDR

Limit address of SAU region37 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION37.NSC

Set NSC for SAU region37 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION38.BADDR

Base address of SAU region38 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION38.ENABLE

Enable SAU region38 at reset.

Type: bool. Default value: 0x0.

SAU_REGION38.LADDR

Limit address of SAU region38 at reset.

Type: int. Default value: 0x0.

SAU_REGION38.NSC

Set NSC for SAU region38 at reset.

Type: bool. Default value: 0x0.

SAU_REGION39.BADDR

Base address of SAU region39 at reset.

Type: int. Default value: 0x0.

SAU_REGION39.ENABLE

Enable SAU region39 at reset.

Type: bool. Default value: 0x0.

SAU_REGION39.LADDR

Limit address of SAU region39 at reset.

Type: int. Default value: 0x0.

SAU_REGION39.NSC

Set NSC for SAU region39 at reset.

Type: bool. Default value: 0x0.

SAU_REGION4.BADDR

Base address of SAU region4 at reset.

Type: int. Default value: 0x0.

SAU_REGION4.ENABLE

Enable SAU region4 at reset.

Type: bool. Default value: 0x0.

SAU_REGION4.LADDR

Limit address of SAU region4 at reset.

Type: int. Default value: 0x0.

SAU_REGION4.NSC

Set NSC for SAU region4 at reset.

Type: bool. Default value: 0x0.

SAU_REGION40.BADDR

Base address of SAU region40 at reset.

Type: int. Default value: 0x0.

SAU_REGION40.ENABLE

Enable SAU region40 at reset.

Type: bool. Default value: 0x0.

SAU_REGION40.LADDR

Limit address of SAU region40 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION40.NSC

Set NSC for SAU region40 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION41.BADDR

Base address of SAU region41 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION41.ENABLE

Enable SAU region41 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION41.LADDR

Limit address of SAU region41 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION41.NSC

Set NSC for SAU region41 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION42.BADDR

Base address of SAU region42 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION42.ENABLE

Enable SAU region42 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION42.LADDR

Limit address of SAU region42 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION42.NSC

Set NSC for SAU region42 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION43.BADDR

Base address of SAU region43 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION43.ENABLE

Enable SAU region43 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION43.LADDR

Limit address of SAU region43 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION43.NSC

Set NSC for SAU region43 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION44.BADDR

Base address of SAU region44 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION44.ENABLE

Enable SAU region44 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION44.LADDR

Limit address of SAU region44 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION44.NSC

Set NSC for SAU region44 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION45.BADDR

Base address of SAU region45 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION45.ENABLE

Enable SAU region45 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION45.LADDR

Limit address of SAU region45 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION45.NSC

Set NSC for SAU region45 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION46.BADDR

Base address of SAU region46 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION46.ENABLE

Enable SAU region46 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION46.LADDR

Limit address of SAU region46 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION46.NSC

Set NSC for SAU region46 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION47.BADDR

Base address of SAU region47 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION47.ENABLE

Enable SAU region47 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION47.LADDR

Limit address of SAU region47 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION47.NSC

Set NSC for SAU region47 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION48.BADDR

Base address of SAU region48 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION48.ENABLE

Enable SAU region48 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION48.LADDR

Limit address of SAU region48 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION48.NSC

Set NSC for SAU region48 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION49.BADDR

Base address of SAU region49 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION49.ENABLE

Enable SAU region49 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION49.LADDR

Limit address of SAU region49 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION49.NSC

Set NSC for SAU region49 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION5.BADDR

Base address of SAU region5 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION5.ENABLE

Enable SAU region5 at reset.
Type: bool. Default value: 0x0.

SAU_REGION5.LADDR

Limit address of SAU region5 at reset.
Type: int. Default value: 0x0.

SAU_REGION5.NSC

Set NSC for SAU region5 at reset.
Type: bool. Default value: 0x0.

SAU_REGION50.BADDR

Base address of SAU region50 at reset.
Type: int. Default value: 0x0.

SAU_REGION50.ENABLE

Enable SAU region50 at reset.
Type: bool. Default value: 0x0.

SAU_REGION50.LADDR

Limit address of SAU region50 at reset.
Type: int. Default value: 0x0.

SAU_REGION50.NSC

Set NSC for SAU region50 at reset.
Type: bool. Default value: 0x0.

SAU_REGION51.BADDR

Base address of SAU region51 at reset.
Type: int. Default value: 0x0.

SAU_REGION51.ENABLE

Enable SAU region51 at reset.
Type: bool. Default value: 0x0.

SAU_REGION51.LADDR

Limit address of SAU region51 at reset.
Type: int. Default value: 0x0.

SAU_REGION51.NSC

Set NSC for SAU region51 at reset.
Type: bool. Default value: 0x0.

SAU_REGION52.BADDR

Base address of SAU region52 at reset.
Type: int. Default value: 0x0.

SAU_REGION52.ENABLE

Enable SAU region52 at reset.
Type: bool. Default value: 0x0.

SAU_REGION52.LADDR

Limit address of SAU region52 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION52.NSC

Set NSC for SAU region52 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION53.BADDR

Base address of SAU region53 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION53.ENABLE

Enable SAU region53 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION53.LADDR

Limit address of SAU region53 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION53.NSC

Set NSC for SAU region53 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION54.BADDR

Base address of SAU region54 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION54.ENABLE

Enable SAU region54 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION54.LADDR

Limit address of SAU region54 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION54.NSC

Set NSC for SAU region54 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION55.BADDR

Base address of SAU region55 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION55.ENABLE

Enable SAU region55 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION55.LADDR

Limit address of SAU region55 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION55.NSC

Set NSC for SAU region55 at reset.
Type: bool. Default value: 0x0.

SAU_REGION56.BADDR

Base address of SAU region56 at reset.
Type: int. Default value: 0x0.

SAU_REGION56.ENABLE

Enable SAU region56 at reset.
Type: bool. Default value: 0x0.

SAU_REGION56.LADDR

Limit address of SAU region56 at reset.
Type: int. Default value: 0x0.

SAU_REGION56.NSC

Set NSC for SAU region56 at reset.
Type: bool. Default value: 0x0.

SAU_REGION57.BADDR

Base address of SAU region57 at reset.
Type: int. Default value: 0x0.

SAU_REGION57.ENABLE

Enable SAU region57 at reset.
Type: bool. Default value: 0x0.

SAU_REGION57.LADDR

Limit address of SAU region57 at reset.
Type: int. Default value: 0x0.

SAU_REGION57.NSC

Set NSC for SAU region57 at reset.
Type: bool. Default value: 0x0.

SAU_REGION58.BADDR

Base address of SAU region58 at reset.
Type: int. Default value: 0x0.

SAU_REGION58.ENABLE

Enable SAU region58 at reset.
Type: bool. Default value: 0x0.

SAU_REGION58.LADDR

Limit address of SAU region58 at reset.
Type: int. Default value: 0x0.

SAU_REGION58.NSC

Set NSC for SAU region58 at reset.
Type: bool. Default value: 0x0.

SAU_REGION59.BADDR

Base address of SAU region59 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION59.ENABLE

Enable SAU region59 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION59.LADDR

Limit address of SAU region59 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION59.NSC

Set NSC for SAU region59 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION6.BADDR

Base address of SAU region6 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION6.ENABLE

Enable SAU region6 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION6.LADDR

Limit address of SAU region6 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION6.NSC

Set NSC for SAU region6 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION60.BADDR

Base address of SAU region60 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION60.ENABLE

Enable SAU region60 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION60.LADDR

Limit address of SAU region60 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION60.NSC

Set NSC for SAU region60 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION61.BADDR

Base address of SAU region61 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION61.ENABLE

Enable SAU region61 at reset.

Type: bool. Default value: 0x0.

SAU_REGION61.LADDR

Limit address of SAU region61 at reset.

Type: int. Default value: 0x0.

SAU_REGION61.NSC

Set NSC for SAU region61 at reset.

Type: bool. Default value: 0x0.

SAU_REGION62.BADDR

Base address of SAU region62 at reset.

Type: int. Default value: 0x0.

SAU_REGION62.ENABLE

Enable SAU region62 at reset.

Type: bool. Default value: 0x0.

SAU_REGION62.LADDR

Limit address of SAU region62 at reset.

Type: int. Default value: 0x0.

SAU_REGION62.NSC

Set NSC for SAU region62 at reset.

Type: bool. Default value: 0x0.

SAU_REGION63.BADDR

Base address of SAU region63 at reset.

Type: int. Default value: 0x0.

SAU_REGION63.ENABLE

Enable SAU region63 at reset.

Type: bool. Default value: 0x0.

SAU_REGION63.LADDR

Limit address of SAU region63 at reset.

Type: int. Default value: 0x0.

SAU_REGION63.NSC

Set NSC for SAU region63 at reset.

Type: bool. Default value: 0x0.

SAU_REGION64.BADDR

Base address of SAU region64 at reset.

Type: int. Default value: 0x0.

SAU_REGION64.ENABLE

Enable SAU region64 at reset.

Type: bool. Default value: 0x0.

SAU_REGION64.LADDR

Limit address of SAU region64 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION64.NSC

Set NSC for SAU region64 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION65.BADDR

Base address of SAU region65 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION65.ENABLE

Enable SAU region65 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION65.LADDR

Limit address of SAU region65 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION65.NSC

Set NSC for SAU region65 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION66.BADDR

Base address of SAU region66 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION66.ENABLE

Enable SAU region66 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION66.LADDR

Limit address of SAU region66 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION66.NSC

Set NSC for SAU region66 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION67.BADDR

Base address of SAU region67 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION67.ENABLE

Enable SAU region67 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION67.LADDR

Limit address of SAU region67 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION67.NSC

Set NSC for SAU region67 at reset.
Type: bool. Default value: 0x0.

SAU_REGION68.BADDR

Base address of SAU region68 at reset.
Type: int. Default value: 0x0.

SAU_REGION68.ENABLE

Enable SAU region68 at reset.
Type: bool. Default value: 0x0.

SAU_REGION68.LADDR

Limit address of SAU region68 at reset.
Type: int. Default value: 0x0.

SAU_REGION68.NSC

Set NSC for SAU region68 at reset.
Type: bool. Default value: 0x0.

SAU_REGION69.BADDR

Base address of SAU region69 at reset.
Type: int. Default value: 0x0.

SAU_REGION69.ENABLE

Enable SAU region69 at reset.
Type: bool. Default value: 0x0.

SAU_REGION69.LADDR

Limit address of SAU region69 at reset.
Type: int. Default value: 0x0.

SAU_REGION69.NSC

Set NSC for SAU region69 at reset.
Type: bool. Default value: 0x0.

SAU_REGION7.BADDR

Base address of SAU region7 at reset.
Type: int. Default value: 0x0.

SAU_REGION7.ENABLE

Enable SAU region7 at reset.
Type: bool. Default value: 0x0.

SAU_REGION7.LADDR

Limit address of SAU region7 at reset.
Type: int. Default value: 0x0.

SAU_REGION7.NSC

Set NSC for SAU region7 at reset.
Type: bool. Default value: 0x0.

SAU_REGION70.BADDR

Base address of SAU region70 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION70.ENABLE

Enable SAU region70 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION70.NSC

Set NSC for SAU region70 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION8.BADDR

Base address of SAU region8 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION8.ENABLE

Enable SAU region8 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION8.LADDR

Limit address of SAU region8 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION8.NSC

Set NSC for SAU region8 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION9.BADDR

Base address of SAU region9 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION9.ENABLE

Enable SAU region9 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION9.LADDR

Limit address of SAU region9 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION9.NSC

Set NSC for SAU region9 at reset.
Type: `bool`. Default value: `0x0`.

SAU_TYPE.SREGION

Number of SAU regions (0 => no SAU).
Type: `int`. Default value: `0x10`.

scheduler_mode

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, 3=ISSCompare) WARNING:

This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `int`. Default value: `0x0`.

semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x20700000`.

semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x20800000`.

semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0x20700000`.

semihosting-Thumb_SVC

T32 SVC number for semihosting.

Type: `int`. Default value: `0xab`.

tcm_cadi_accesses_are_physical

CADI accesses to TCMs ignore any alias regions configured (for validation platforms).

Type: `bool`. Default value: `0x0`.

use_architectural_names

Enable using architectural names for registers LR/PSP/MSP instead of model-specific names.

Type: `bool`. Default value: `0x0`.

vector_fetch_busfault_sets_HFSR_FORCED

Only v8.0M. Set HFSR.FORCED when a vector table read generates a HardFault.

Type: `bool`. Default value: `0x0`.

vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

VTOR_NS

NonSecure Vector Table Offset Register is writeable.

Type: `bool`. Default value: `0x1`.

VTOR_NS_MASK

Non-Secure VTOR write mask.

Type: `int`. Default value: `0xffffffff80`.

VTOR_S

Secure Vector Table Offset Register is writeable.

Type: `bool`. Default value: `0x1`.

VTOR_S_MASK

Secure VTOR write mask.

Type: `int`. Default value: `0xffffffff80`.

wr_ns_bus_err_behave

External write aborts in nonsecure domain 0:ignored, 1:precise, 2:imprecise, 3=imprecise except SO.

Type: `int`. Default value: `0x3`.

wr_s_bus_err_behave

External write aborts in secure domain 0:ignored, 1:precise, 2:imprecise, 3=imprecise except SO.

Type: `int`. Default value: `0x3`.

4.5.4 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 4-80: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA5CT contains the following CADI targets:

- ARM_Cortex-A5UP

- Cluster_ARM_Cortex-A5UP
- PVCache
- TlbCadi

ARMCortexA5CT contains the following MTI components:

- [ARM_Cortex-A5](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

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 SCTLRR 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 4-81: 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 ARM_Cortex-A5UP

cpu0.ase-present

Set whether model has NEON support.

Type: bool. Default value: 0x1.

cpu0.CFGEND

Initialize to BE8 endianness.

Type: bool. Default value: 0x0.

cpu0.CFGNMFI

Enable nonmaskable FIQ interrupts on startup.

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: bool. Default value: 0x0.

cpu0.dcache-size

Set D-cache size in bytes.

Type: int. Default value: 0x8000.

cpu0.icache-size

Set I-cache size in bytes.

Type: int. Default value: 0x8000.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.POWERCTLI

Default power control state for processor.

Type: int. Default value: 0x0.

cpu0.semihosting-ARM_HLT

ARM HLT number for semihosting.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

ARM SVC number for semihosting.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: int. Default value: 0xf000000.

cpu0.semihosting-hlt-enable

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: bool. Default value: 0x0.

cpu0.semihosting-prefix

Prefix semihosting output with target instance name.

Type: bool. Default value: 0x0.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: int. Default value: 0xf000000.

cpu0.semihosting-Thumb_HLT

Thumb HLT number for semihosting.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

Thumb SVC number for semihosting.

Type: `int`. Default value: `0xab`.

cpu0.TEINIT

T32 exception enable. The default has exceptions including reset handled in A32 state.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-enable_at_reset

Enable coprocessor access and VFP at reset.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

cpu0.VINITHI

Initialize with high vectors enabled.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A5UP**CLUSTER_ID**

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-hit_latency

D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

device-accurate-tlb

Specify whether all TLBs are modeled.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

4.5.5 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 4-82: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA5MPx1CT contains the following CADI targets:

- `ARM_Cortex-A5MP`
- `Cluster_ARM_Cortex-A5MP`
- `PVCache`
- `TlbCadi`

ARMCortexA5MPx1CT contains the following MTI components:

- [ARM_Cortex-A5MP](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)

- [TLB](#)

About ARMCortexA5MPx1CT

- The following components also exist:
 - ARMCortexA5MPx2CT.
 - ARMCortexA5MPx4CT.

The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-3).

- The `ase-present` and `vfp-present` parameters configure the synthesis options:

vfp present and ase present

Neon™ and VFPv3-D32 are supported.

vfp present and ase not present

VFPv3-D16 is supported.

vfp not present and ase present

Illegal. Forces `vfp-present` to true so model has Neon™ and VFPv3-D32 support.

vfp not present and ase not present

Model has neither Neon™ nor VFPv3-D32 support.

- Specify `false` for the `device-accurate-tlb` parameter to enable modeling a different number of TLBs if this improves simulation performance. In this case, the simulation is architecturally-accurate, but not device-accurate. Architectural accuracy is almost always sufficient. Specify `true` if device accuracy is required.
- If you are using the ARMCortexA5MPx n CT component in a VE platform model, the `PERIPHBASE` parameter is set automatically to `0x2C000000` and is not visible in the parameter list.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to `false` in production systems.

Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- There is a single memory port combining instruction, data, DMA and peripheral access.
- The GIC does not respect the `CFGSDISABLE` signal. This leads to some registers being accessible when they must not be.
- The SCU enable bit is ignored. The SCU is always enabled.
- The SCU ignores the invalidate all register.
- The Broadcast TLB or cache operations in this model do not cause other cores in the cluster that are asleep because of *Wait For Interrupt* (WFI) to wake up.
- The RR bit in the SCTLR is ignored.
- The Power Control Register in the system control coprocessor is implemented but writing to it does not change the behavior of the model.
- When modeling the SCU, coherency operations are represented by a memory write followed by a read to refill from memory, rather than using cache-to-cache transfers.

- This component implements L1 cache as architecturally defined, but does not implement L2 cache. If you require an L2 cache you can add a PL310 Level 2 Cache Controller component.

Ports for ARMCortexA5MPx1CT

Table 4-83: 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.
pwrctl[1]	Value	Slave	This port sets reset value for scu CPU status register.
pwrctl0[1]	Value	Master	This port sends scu CPU status register bits.
reset[1]	Signal	Slave	Raising this signal will put the core into reset mode.
scureset	Signal	Slave	This signal resets SCU.
smpnamp[1]	Signal	Master	This signals AMP or SMP mode for each Cortex-A5 processor.
standbywfe[1]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[1]	Signal	Master	This signal indicates if a core is in WFI state.
teinit[1]	Signal	Slave	This signal provides default exception handling state.
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.

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).
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-A5MP

cpu0.ase-present

Set whether model has NEON support.
Type: bool. Default value: 0x1.

cpu0.CFGEND

Initialize to BE8 endianness.
Type: bool. Default value: 0x0.

cpu0.CFGNMFI

Enable nonmaskable FIQ interrupts on startup.
Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.
Type: bool. Default value: 0x0.

cpu0.dcache-size

Set D-cache size in bytes.
Type: int. Default value: 0x8000.

cpu0.icache-size

Set I-cache size in bytes.
Type: int. Default value: 0x8000.

cpu0.min_sync_level

force minimum syncLevel (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll).
Type: int. Default value: 0x0.

cpu0.POWERCTLI

Default power control state for processor.
Type: int. Default value: 0x0.

cpu0.semihosting-ARM_HLT

ARM HLT number for semihosting.
Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

ARM SVC number for semihosting.
Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting SVC calls.
Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-hlt-enable

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: `bool`. Default value: `0x0`.

cpu0.semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-Thumb_HLT

Thumb HLT number for semihosting.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

Thumb SVC number for semihosting.

Type: `int`. Default value: `0xab`.

cpu0.SMPnAMP

Set whether the processor is part of a coherent domain.

Type: `bool`. Default value: `0x0`.

cpu0.TEINIT

T32 exception enable. The default has exceptions including reset handled in A32 state.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-enable_at_reset

Enable coprocessor access and VFP at reset.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.
Type: `bool`. Default value: `0x1`.

cpu0.VINITHI

Initialize with high vectors enabled.
Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A5MP**CFGSDISABLE**

Disable some accesses to GIC registers.
Type: `bool`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.
Type: `int`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

dcache-hit_latency

D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-read_access_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

`dcache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-state_modelled`

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

`dcache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`device-accurate-tlb`

Specify whether all TLBs are modeled.

Type: `bool`. Default value: `0x0`.

`dic-spi_count`

Number of shared peripheral interrupts implemented.

Type: `int`. Default value: `0x40`.

`FILTEREN`

Enable filtering of accesses through `pvbus_m0`.

Type: `bool`. Default value: `0x0`.

`FILTEREND`

End of region filtered to `pvbus_m0`.

Type: `int`. Default value: `0x0`.

`FILTERSTART`

Base of region filtered to `pvbus_m0`.

Type: `int`. Default value: `0x0`.

`icache-hit_latency`

I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`icache-maintenance_latency`

I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

PERIPHBASE

Base address of peripheral memory space.

Type: `int`. Default value: `0x13080000`.

4.5.6 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 4-84: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA7x1CT contains the following CADI targets:

- ARM_Cortex-A7
- Cluster_ARM_Cortex-A7
- PVCache
- TlbCadi

ARMCortexA7x1CT contains the following MTI components:

- [ARM_Cortex-A7](#)
- [AsyncCacheFlushUnit](#)
- [GICv2](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

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 `PERIPHBASE` parameter is set automatically to `0x2C000000` and is not visible in the parameter list.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to `false` in production systems.

ACE limitation

AXI Coherency Extensions (ACE) are extensions to AXI4 that support system-level cache-coherency between multiple clusters. The ACE cache models in the Cortex®-A15 and the 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/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 4-85: 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 ARM_Cortex-A7

cpu0.ase-present

Set whether CT model has been built with NEON support.

Type: bool. Default value: 0x1.

cpu0.CFGEND

Initialize to BE8 endianness.

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: bool. Default value: 0x0.

cpu0.DBGROMADDR

This value is used to initialize the CP15 DBGDRAR register. Bits[39:12] of this register specify the ROM table physical address.

Type: int. Default value: 0x12000003.

cpu0.DBGROMADDRV

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

Type: bool. Default value: 0x1.

cpu0.DBGSELFADDR

This value is used to initialize the CP15 DBGDSAR register. Bits[39:17] of this register specify the ROM table physical address.

Type: int. Default value: 0x10003.

cpu0.DBGSELFADDRV

If true this sets bits[1:0] of the CP15 DBGDSAR to indicate that the address is valid.

Type: bool. Default value: 0x1.

cpu0.l1_dcache-size

Size of L1 D-cache.

Type: int. Default value: 0x8000.

cpu0.l1_icode-size

Size of L1 I-cache.

Type: int. Default value: 0x8000.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.semihosting-ARM_HLT

ARM HLT number for semihosting.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

ARM SVC number for semihosting.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: int. Default value: 0xf000000.

cpu0.semihosting-hlt-enable

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: bool. Default value: 0x0.

cpu0.semihosting-prefix

Prefix semihosting output with target instance name.

Type: bool. Default value: 0x0.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: int. Default value: 0xf000000.

cpu0.semihosting-Thumb_HLT

Thumb HLT number for semihosting.

Type: int. Default value: 0x3c.

cpu0.semihosting-Thumb_SVC

Thumb SVC number for semihosting.

Type: int. Default value: 0xab.

cpu0.TEINIT

T32 exception enable. The default has exceptions including reset handled in A32 state.

Type: bool. Default value: 0x0.

cpu0.vfp-enable_at_reset

Enable coprocessor access and VFP at reset.

Type: bool. Default value: 0x0.

cpu0.vfp-present

Set whether CT model has been built with VFP support.

Type: bool. Default value: 0x1.

cpu0.VINITHI

Initialize with high vectors enabled.

Type: bool. Default value: 0x0.

Parameters for Cluster_ARM_Cortex-A7**BROADCASTCHEMAINT**

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

Type: bool. Default value: 0x1.

BROADCASTINNER

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

Type: bool. Default value: 0x1.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

CFGSDISABLE

Disable some accesses to GIC registers.

Type: `bool`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dic-spi_count

Number of shared peripheral interrupts implemented.

Type: `int`. Default value: `0x40`.

disable_periph_decoder

Disable memory mapped access to gic system registers.

Type: `bool`. Default value: `0x0`.

internal_vgic

Configures whether the model of the processor contains a VGIC.

Type: `bool`. Default value: `0x1`.

l1_dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l1_dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l1_dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

`l1_dcache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`l1_dcache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`l1_dcache-state_modelled`

Set whether L1 D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

`l1_dcache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`l1_dcache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`l1_icache-hit_latency`

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`l1_icache-maintenance_latency`

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`l1_icache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

l1_icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-state_modelled

Set whether L1 I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l2_cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2_cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2_cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-size

Set L2 cache size in bytes.

Type: `int`. Default value: `0x80000`.

l2_cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2_cache-state_modelled

Set whether L2 cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l2_cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

PERIPHBASE

Base address of peripheral memory space.

Type: `int`. Default value: `0x13080000`.

4.5.7 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 4-86: IP revisions support

Revision	Quality level
r2p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA8CT contains the following CADI targets:

- ARM_Cortex-A8
- PVCache
- TlbCadi

ARMCortexA8CT contains the following MTI components:

- [ARM_Cortex-A8](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

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_dcachestate_modelled` parameter is `true`, then `l2_cachestate_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 4-87: 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.
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
reset	Signal	Slave	Raising this signal will put the core into reset mode.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.
vinithi	Signal	Slave	Configure high vectors after a reset.

Parameters for ARM_Cortex-A8

CFGEND0

Initialize to BE8 endianness.
Type: bool. Default value: 0x0.

CFGNMFI

Enable nonmaskable FIQ interrupts on startup.
Type: bool. Default value: 0x0.

CFGTE

Initialize to take exceptions in T32 state. Model starts in T32 state.
Type: bool. Default value: 0x0.

CP15SDISABLE

Initialize to disable access to some CP15 registers.
Type: bool. Default value: 0x0.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).
Type: int. Default value: 0x1.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).
Type: int. Default value: 0x1.

device-accurate-tlb

Specify whether all TLBs are modeled.

Type: `bool`. Default value: `0x0`.

implements_vfp

Set whether the model has been built with VFP and NEON support.

Type: `bool`. Default value: `0x1`.

l1_dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l1_dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l1_dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-size

Set L1 D-cache size in bytes.

Type: `int`. Default value: `0x8000`.

l1_dcache-state_modelled

Include Level 1 data cache state model.

Type: `bool`. Default value: `0x0`.

l1_dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l1_icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-size

Set L1 I-cache size in bytes.

Type: `int`. Default value: `0x8000`.

l1_icache-state_modelled

Include Level 1 instruction cache state model.

Type: `bool`. Default value: `0x0`.

l2_cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2_cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2_cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-size

Set L2 cache size in bytes.

Type: `int`. Default value: `0x40000`.

l2_cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-state_modelled

Include unified Level 2 cache state model.

Type: `bool`. Default value: `0x0`.

l2_cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

master_id

Master ID presented in bus transactions.

Type: `int`. Default value: `0x0`.

min_sync_level

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

Type: `int`. Default value: `0x0`.

semihosting-ARM_HLT

ARM HLT number for semihosting.
Type: `int`. Default value: `0xf000`.

semihosting-ARM_SVC

ARM SVC number for semihosting.
Type: `int`. Default value: `0x123456`.

semihosting-cmd_line

Command line available to semihosting SVC calls.
Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.
Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.
Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.
Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.
Type: `int`. Default value: `0xf0000000`.

semihosting-hlt-enable

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.
Type: `bool`. Default value: `0x0`.

semihosting-prefix

Prefix semihosting output with target instance name.
Type: `bool`. Default value: `0x0`.

semihosting-stack_base

Virtual address of base of descending stack.
Type: `int`. Default value: `0x10000000`.

semihosting-stack_limit

Virtual address of stack limit.
Type: `int`. Default value: `0xf0000000`.

semihosting-Thumb_HLT

Thumb HLT number for semihosting.
Type: `int`. Default value: `0x3c`.

semihosting-Thumb_SVC

Thumb SVC number for semihosting.
Type: `int`. Default value: `0xab`.

- siliconID**
Value as read by the system coprocessor siliconID register.
Type: int. Default value: 0x41000000.
- vfp-enable_at_reset**
Enable coprocessor access and VFP at reset.
Type: bool. Default value: 0x0.
- VINITHI**
Initialize with high vectors enabled.
Type: bool. Default value: 0x0.

4.5.8 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 4-88: IP revisions support

Revision	Quality level
r3p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA9MPx1CT contains the following CADI targets:

- ARM_Cortex-A9MP
- Cluster_ARM_Cortex-A9MP
- PVCache
- TlbCadi

ARMCortexA9MPx1CT contains the following MTI components:

- [ARM_Cortex-A9MP](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

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 4-89: Ports

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgnmfi[4]	Signal	Slave	This signal disables FIQ mask in CPSR.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
filteren	Signal	Slave	This signal enables filtering of address ranges between master bus ports.
filterend	Value	Slave	This port sets end of region mapped to pvbus_m1.
filterstart	Value	Slave	This port sets start of region mapped to pvbus_m1.
fiq[4]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
fiqout[4]	Signal	Master	This signal exports internal VGIC FIQ signal to the CPU..
ints[224]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
irq[4]	Signal	Slave	This signal drives the CPU's interrupt handling.
irqout[4]	Signal	Master	This signal exports internal VGIC IRQ signal to the CPU.
periphbase	Value	Slave	This port sets the base address of private peripheral region.
periphclk_in	ClockSignal	Slave	The timer and the watchdog take need a clk that is scaled down at least by factor of two.
periphreset	Signal	Slave	This signal resets timer and interrupt controller.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
pvbus_m0	PVBus	Master	AXI master 0 bus master channel.
pvbus_m1	PVBus	Master	AXI master 1 bus master channel.

Name	Protocol	Type	Description
pwrctl[4]	Value	Slave	This port sets reset value for scu CPU status register.
pwrctl[4]	Value	Master	This port sends scu CPU status register bits.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
scureset	Signal	Slave	This signal resets SCU.
smpnamp[4]	Signal	Master	This signals AMP or SMP mode for each Cortex-A9 processor.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
teinit[4]	Signal	Slave	This signal provides default exception handling state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
wdreset[4]	Signal	Slave	This signal resets individual watchdog.
wdresetreq[4]	Signal	Master	This signal resets rest of the CA9MP system.

Parameters for ARM_Cortex-A9MP

cpu0.ase-present

Set whether model has NEON support.

Type: bool. Default value: 0x1.

cpu0.CFGEND

Initialize to BE8 endianness.

Type: bool. Default value: 0x0.

cpu0.CFGNMFI

Enable nonmaskable FIQ interrupts on startup.

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: bool. Default value: 0x0.

cpu0.dcache-size

Set D-cache size in bytes.

Type: int. Default value: 0x8000.

cpu0.icache-size

Set I-cache size in bytes.

Type: int. Default value: 0x8000.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.POWERCTLI

Default power control state for processor.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-ARM_HLT`

ARM HLT number for semihosting.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

ARM SVC number for semihosting.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-hlt-enable`

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: `bool`. Default value: `0x0`.

`cpu0.semihosting-prefix`

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-Thumb_HLT`

Thumb HLT number for semihosting.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

Thumb SVC number for semihosting.

Type: `int`. Default value: `0xab`.

`cpu0.SMPnAMP`

Set whether the processor is part of a coherent domain.

Type: `bool`. Default value: `0x0`.

`cpu0.TEINIT`

T32 exception enable. The default has exceptions including reset handled in A32 state.

Type: `bool`. Default value: `0x0`.

`cpu0.vfp-enable_at_reset`

Enable coprocessor access and VFP at reset.

Type: `bool`. Default value: `0x0`.

`cpu0.vfp-present`

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

`cpu0.VINITHI`

Initialize with high vectors enabled.

Type: `bool`. Default value: `0x0`.

Parameters for `Cluster_ARM_Cortex-A9MP`

`CFGSDISABLE`

Disable some accesses to GIC registers.

Type: `bool`. Default value: `0x0`.

`CLUSTER_ID`

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

`cpi_div`

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

`cpi_mul`

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

`dcache-hit_latency`

D-cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`dcache-maintenance_latency`

D-cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`dcache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

device-accurate-tlb

Specify whether all TLBs are modeled.

Type: `bool`. Default value: `0x0`.

dic-spi_count

Number of shared peripheral interrupts implemented.

Type: `int`. Default value: `0x40`.

FILTEREN

Enable filtering of accesses through `pvbus_m0`.

Type: `bool`. Default value: `0x0`.

FILTEREND

End of region filtered to `pvbus_m0`.

Type: `int`. Default value: `0x0`.

FILTERSTART

Base of region filtered to `pvbus_m0`.

Type: `int`. Default value: `0x0`.

icache-hit_latency

I-cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

I-cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

PERIPHBASE

Base address of peripheral memory space.

Type: `int`. Default value: `0x13080000`.

4.5.9 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 4-90: IP revisions support

Revision	Quality level
r3p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA9UPCT contains the following CADI targets:

- ARM_Cortex-A9UP
- Cluster_ARM_Cortex-A9UP
- PVCache
- TlbCadi

ARMCortexA9UPCT contains the following MTI components:

- [ARM_Cortex-A9UP](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

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 4-91: 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.
<code>cp15sdisable[1]</code>	Signal	Slave	This signal disables write access to some system control processor registers.
<code>event</code>	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
<code>fiq[1]</code>	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
<code>irq[1]</code>	Signal	Slave	This signal drives the CPU's interrupt handling.
<code>pmuirq[1]</code>	Signal	Master	Interrupt signal from performance monitoring unit.
<code>pvbus_m0</code>	PVBus	Master	The core will generate bus requests on this port.
<code>reset[1]</code>	Signal	Slave	Raising this signal will put the core into reset mode.

Name	Protocol	Type	Description
standbywfe[1]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[1]	Signal	Master	This signal indicates if a core is in WFI state.
teinit[1]	Signal	Slave	This signal provides default exception handling state.
ticks[1]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vinithi[1]	Signal	Slave	This signal controls of the location of the exception vectors at reset.

Parameters for ARM_Cortex-A9UP

cpu0.ase-present

Set whether model has NEON support.

Type: bool. Default value: 0x1.

cpu0.CFGEND

Initialize to BE8 endianness.

Type: bool. Default value: 0x0.

cpu0.CFGNMFI

Enable nonmaskable FIQ interrupts on startup.

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: bool. Default value: 0x0.

cpu0.dcache-size

Set D-cache size in bytes.

Type: int. Default value: 0x8000.

cpu0.icache-size

Set I-cache size in bytes.

Type: int. Default value: 0x8000.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.POWERCTLI

Default power control state for processor.

Type: int. Default value: 0x0.

cpu0.semihosting-ARM_HLT

ARM HLT number for semihosting.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

ARM SVC number for semihosting.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-hlt-enable`

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: `bool`. Default value: `0x0`.

`cpu0.semihosting-prefix`

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-Thumb_HLT`

Thumb HLT number for semihosting.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

Thumb SVC number for semihosting.

Type: `int`. Default value: `0xab`.

`cpu0.TEINIT`

T32 exception enable. The default has exceptions including reset handled in A32 state.

Type: `bool`. Default value: `0x0`.

`cpu0.vfp-enable_at_reset`

Enable coprocessor access and VFP at reset.

Type: `bool`. Default value: `0x0`.

`cpu0.vfp-present`

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

`cpu0.VINITHI`

Initialize with high vectors enabled.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A9UP

`CLUSTER_ID`

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

`cpi_div`

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

`cpi_mul`

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

`dcache-hit_latency`

D-cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`dcache-maintenance_latency`

D-cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`dcache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-state_modelled`

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

device-accurate-tlb

Specify whether all TLBs are modeled.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

I-cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

I-cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

4.5.10 ARM Cortex A15x1CT

ARM Cortex A15x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-92: IP revisions support

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM Cortex A15x1CT contains the following CADI targets:

- ARM_Cortex-A15
- Cluster_ARM_Cortex-A15
- PVCache
- TlbCadi

ARM Cortex A15x1CT contains the following MTI components:

- [ARM_Cortex-A15](#)
- [AsyncCacheFlushUnit](#)
- [GICv2](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

About ARM Cortex A15x1CT

- The following components also exist:
 - ARM Cortex A15x2CT.
 - ARM Cortex A15x3CT.
 - ARM Cortex A15x4CT.

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 `ARMCortexA15xCT` 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 Cortex®-A15 and the 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/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 4-93: Ports

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
broadcastcachemaint	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	Signal	Slave	Enable broadcasting of Inner Shareable transactions.
broadcastouter	Signal	Slave	Enable broadcasting of Outer Shareable transactions.
cfgend[4]	Signal	Slave	This signal controls the SCTLR.EE bit.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	This port sets the value in the CLUSTERID field (bits[11:8]) of the MPIDR.
CNTHPIRQ[4]	Signal	Master	Outputs of the generic timers.
CNTPNSIRQ[4]	Signal	Master	Outputs of the generic timers.
CNTPSIRQ[4]	Signal	Master	Outputs of the generic timers.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Outputs of the generic timers.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some secure system control processor registers.
cpuporeset[4]	Signal	Slave	Signal initializes all processor logic including NEON, VFP, Debug, PTM, breakpoint and watchpoint.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
fiqout[4]	Signal	Master	This signal exports internal VGIC FIQ signal to the CPU..
irq[4]	Signal	Slave	This signal drives the CPU's interrupt handling.
irqout[4]	Signal	Master	This signal exports internal VGIC IRQ signal to the CPU.
irqs[224]	Signal	Slave	These signals drive the CPU's interrupt controller interrupt lines.
l2reset	Signal	Slave	This signal resets the shared L2 memory system, interrupt controller and timer logic.
periphbase	Value_64	Slave	This port sets the base address of the private peripheral region.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Signal initializes the shared Debug APB, CTI and CTM logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
standbywfi12	Signal	Master	Indicate that all the individual processors and the L2 memory system are in a WFI state.
teinit[4]	Signal	Slave	This signal enables Thumb exceptions (controls the SCTLR.TE bit).
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.

Name	Protocol	Type	Description
vfiq[4]	Signal	Slave	Virtual FIQ inputs. Note that the fiq pins are wired directly to the core if there is no internal VGIC. If there is an internal VGIC then these are ignored.
vinithi[4]	Signal	Slave	This signal controls the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtual IRQ inputs. Note that the irq pins are wired directly to the core if there is no internal VGIC. If there is an internal VGIC then these are ignored.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).

Parameters for ARM_Cortex-A15

cpu0.ase-present

Set whether CT model has been built with NEON support.

Type: bool. Default value: 0x1.

cpu0.CFGEND

Initialize to BE8 endianness.

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: bool. Default value: 0x0.

cpu0.DBGROMADDR

This value is used to initialize the CP15 DBGDRAR register. Bits[39:12] of this register specify the ROM table physical address.

Type: int. Default value: 0x12000003.

cpu0.DBGROMADDRV

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

Type: bool. Default value: 0x1.

cpu0.DBGSELFADDR

This value is used to initialize the CP15 DBGDSAR register. Bits[39:17] of this register specify the ROM table physical address.

Type: int. Default value: 0x10003.

cpu0.DBGSELFADDRV

If true, this sets bits[1:0] of the CP15 DBGDSAR to indicate that the address is valid.

Type: bool. Default value: 0x1.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.semihosting-ARM_HLT

ARM HLT number for semihosting.

Type: int. Default value: 0x1000.

cpu0.semihosting-ARM_SVC

ARM SVC number for semihosting.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-hlt-enable

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: `bool`. Default value: `0x0`.

cpu0.semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-Thumb_HLT

Thumb HLT number for semihosting.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

Thumb SVC number for semihosting.

Type: `int`. Default value: `0xab`.

cpu0.TEINIT

T32 exception enable. The default has exceptions including reset handled in A32 state.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-enable_at_reset

Enable coprocessor access and VFP at reset.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether CT model has been built with VFP support.

Type: `bool`. Default value: `0x1`.

cpu0.VINITHI

Initialize with high vectors enabled.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A15**BROADCASTCACHEMAINT**

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

Type: `bool`. Default value: `0x1`.

BROADCASTINNER

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

CFGSDISABLE

Disable some accesses to GIC registers.

Type: `bool`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dic-spi_count

Number of shared peripheral interrupts implemented.

Type: `int`. Default value: `0x40`.

disable_periph_decoder

Disable memory mapped access to gic system registers.

Type: `bool`. Default value: `0x0`.

IMINLN

Instruction cache minimum line size: false=32 bytes, true=64 bytes.

Type: `bool`. Default value: `0x1`.

internal_vgic

Configures whether the model of the processor contains a Virtualized Generic Interrupt Controller (VGIC).

Type: `bool`. Default value: `0x1`.

l1_dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l1_dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l1_dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-state_modelled

Set whether L1 D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l1_dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l1_icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l1_icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-state_modelled

Set whether L1 I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l2-data-slice

L2 data RAM slice.

Type: `int`. Default value: `0x0`.

l2-tag-slice

L2 tag RAM slice.

Type: `int`. Default value: `0x0`.

l2_cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2_cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2_cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-size

Set L2 cache size in bytes.

Type: `int`. Default value: `0x80000`.

l2_cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2_cache-state_modelled

Set whether L2 cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l2_cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

PERIPHBASE

Base address of peripheral memory space.

Type: `int`. Default value: `0x13080000`.

4.5.11 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 4-94: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA17x1CT contains the following CADI targets:

- `ARM_Cortex-A17`
- `Cluster_ARM_Cortex-A17`
- `PVCache`
- `TlbCadi`

ARMCortexA17x1CT contains the following MTI components:

- [ARM_Cortex-A17](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

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 4-95: 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 SCTL.R.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.

Name	Protocol	Type	Description
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Outputs of the generic timers.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some secure system control processor registers.
cpuporeset[4]	Signal	Slave	Signal initializes all processor logic including NEON, VFP, Debug, PTM, breakpoint and watchpoint.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
irq[4]	Signal	Slave	This signal drives the CPU's interrupt handling.
l2flushdone	Signal	Master	Flush of L2 memory system complete.
l2flushreq	Signal	Slave	Request flush of L2 memory system.
l2reset	Signal	Slave	This signal resets the shared L2 memory system and timer logic.
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.
pvbuse_m0	PVBus	Master	The core will generate bus requests on this port.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
standbywfil2	Signal	Master	Indicate that all the individual processors and the L2 memory system are in a WFI state.
teinit[4]	Signal	Slave	This signal enables Thumb exceptions (controls the SCTLR.TE bit).
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vfiq[4]	Signal	Slave	Virtual FIQ inputs.
vinithi[4]	Signal	Slave	This signal controls the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtual IRQ inputs.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).

Parameters for ARM_Cortex-A17

cpu0.ase-present

Set whether CT model has been built with NEON support.
Type: boo1. Default value: 0x1.

cpu0.CFGEND

Initialize to BE8 endianness.
Type: boo1. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.
Type: boo1. Default value: 0x0.

cpu0.DBGROMADDR

This value is used to initialize the CP15 DBGDRAR register. Bits[39:12] of this register specify the ROM table physical address.

Type: `int`. Default value: `0x12000003`.

cpu0.DBGROMADDRV

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

Type: `bool`. Default value: `0x1`.

cpu0.DBGSELFADDR

This value is used to initialize the CP15 DBGDSAR register. Bits[39:17] of this register specify the ROM table physical address.

Type: `int`. Default value: `0x10003`.

cpu0.DBGSELFADDRV

If true, this sets bits[1:0] of the CP15 DBGDSAR to indicate that the address is valid.

Type: `bool`. Default value: `0x1`.

cpu0.l1_icache-size

Size of L1 I-cache.

Type: `int`. Default value: `0x8000`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.semihosting-ARM_HLT

ARM HLT number for semihosting.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

ARM SVC number for semihosting.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-hlt-enable

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: `bool`. Default value: `0x0`.

cpu0.semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-Thumb_HLT

Thumb HLT number for semihosting.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

Thumb SVC number for semihosting.

Type: `int`. Default value: `0xab`.

cpu0.TEINIT

T32 exception enable. The default has exceptions including reset handled in A32 state.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-enable_at_reset

Enable coprocessor access and VFP at reset.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether CT model has been built with VFP support.

Type: `bool`. Default value: `0x1`.

cpu0.VINITHI

Initialize with high vectors enabled.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A17

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTINNER

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

CFGADDRFILTEENDNS

Peripheral port NS end address.

Type: `int`. Default value: `0x0`.

CFGADDRFILTEENDS

Peripheral port S end address.

Type: `int`. Default value: `0x0`.

CFGADDRFILTEENNS

Peripheral port NS address filtering enabled.

Type: `bool`. Default value: `0x0`.

CFGADDRFILTEENS

Peripheral port S address filtering enabled.

Type: `bool`. Default value: `0x0`.

CFGADDRFILTESTARTNS

Peripheral port NS start address.

Type: `int`. Default value: `0x0`.

CFGADDRFILTESTARTS

Peripheral port S start address.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

IMINLN

Instruction cache minimum line size: `false`=32 bytes, `true`=64 bytes.

Type: `bool`. Default value: `0x1`.

l1_dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l1_dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l1_dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-state_modelled

Set whether L1 D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l1_dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l1_dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l1_icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l1_icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l1_icache-state_modelled

Set whether L1 I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l2-data-slice

L2 data RAM slice.

Type: `int`. Default value: `0x0`.

l2-tag-slice

L2 tag RAM slice.

Type: `int`. Default value: `0x0`.

l2_cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2_cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2_cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-size

Set L2 cache size in bytes.

Type: `int`. Default value: `0x40000`.

l2_cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2_cache-state_modelled

Set whether L2 cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l2_cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2_cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

4.5.12 ARM Cortex A32x1CT

ARM Cortex A32x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-96: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM Cortex A32x1CT contains the following CADI targets:

- ARM_Cortex-A32
- Cluster_ARM_Cortex-A32
- PVCache
- TlbCadi

ARM Cortex A32x1CT contains the following MTI components:

- [ARM_Cortex-A32](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

About ARM Cortex A32x1CT

- The following components also exist:
 - ARM Cortex A32x2CT.
 - ARM Cortex A32x3CT.
 - ARM Cortex A32x4CT.

The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-3).

- 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 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 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](#).
- 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` Of `memorymapped_debug_s`.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to `false` in production systems.
- ECC and parity schemes are hardware-specific so are not supported.

Ports for ARMCortexA32x1CT

Table 4-97: 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.
<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.
cp15sdisable2[4]	Signal	Slave	-
cpuporeset[4]	Signal	Slave	CPU power on reset.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port
ctidbgirq[4]	Signal	Master	Cross trigger matrix port.
dbgack[4]	Signal	Master	External debug interface.
dbgen[4]	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Processor powerup request.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
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.

Name	Protocol	Type	Description
romaddrv	Signal	Slave	Debug ROM base address valid.
sei[4]	Signal	Slave	Per core System Error physical pins.
smpen[4]	Signal	Master	This signals AMP or SMP mode for each core.
spiden[4]	Signal	Slave	External debug interface.
spniden[4]	Signal	Slave	External debug interface.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state
standbywfil2	Signal	Master	This signal indicated all cores and L2 are idles and in low power state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-A32

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CFGTE

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

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE2

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: `bool`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF enable_trace_special_hlt_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: int. Default value: 0xf000.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool. Default value: 0x0.

cpu0.vfp-present

Set whether the model has VFP support.

Type: bool. Default value: 0x1.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: bool. Default value: 0x0.

Parameters for Cluster_ARM_Cortex-A32**BROADCASTCACHEMAINT**

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

Type: bool. Default value: 0x1.

BROADCASTINNER

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

Type: bool. Default value: 0x1.

BROADCASTOUTER

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

Type: bool. Default value: 0x1.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: int. Default value: 0x0.

CLUSTER_ID

Processor cluster ID value.

Type: int. Default value: 0x0.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: int. Default value: 0x1.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

DBGROMADDR

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

Type: `int`. Default value: `0x22000000`.

DBGROMADDRV

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

Type: `bool`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

icache-read_latency

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.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

l2cache-miss_latency

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.

Type: `int`. Default value: `0x0`.

l2cache-read_access_latency

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.

Type: int. Default value: 0x0.

l2cache-read_latency

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.

Type: int. Default value: 0x0.

l2cache-size

L2 Cache size in bytes.

Type: int. Default value: 0x80000.

l2cache-snoop_data_transfer_latency

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.

Type: int. Default value: 0x0.

l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

l2cache-write_access_latency

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.

Type: int. Default value: 0x0.

l2cache-write_latency

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.

Type: int. Default value: 0x0.

PERIPHBASE

Base address of peripheral memory space.

Type: int. Default value: 0x13080000.

ptw_latency

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

Type: int. Default value: 0x0.

tlb_latency

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

Type: int. Default value: 0x0.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.13 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 4-98: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA34x1CT contains the following CADI targets:

- ARM_Cortex-A34
- Cluster_ARM_Cortex-A34
- PVCache
- TlbCadi

ARMCortexA34x1CT contains the following MTI components:

- [ARM_Cortex-A34](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

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 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](#).
- 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`.
- 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.
- The following components also exist:
 - `ARMCortexA34x2CT`
 - `ARMCortexA34x3CT`
 - `ARMCortexA34x4CT`

The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-3).

Ports for ARMCortexA34x1CT

Table 4-99: 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.
<code>broadcastouter</code>	Signal	Slave	ACE 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 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

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.
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.
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.

Name	Protocol	Type	Description
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
rei[4]	Signal	Slave	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.
vfirq[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-A34

cpu0.AA64nAA32

Register width configuration at reset. 0, AArch32. 1, AArch64.
Type: bool. Default value: 0x1.

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CFGTE

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

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE2

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.max_code_cache_mb

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).

Type: int. Default value: 0x100.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: int. Default value: 0xf000000.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: int. Default value: 0x3c.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: int. Default value: 0xab.

cpu0.trace_special_hlt_imm16

For this HLT number, IF enable_trace_special_hlt_imm16=true, skip performing usual HLT execution but call MTI trace if registered.
Type: int. Default value: 0xf000.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: bool. Default value: 0x0.

cpu0.vfp-present

Set whether the model has VFP support.
Type: bool. Default value: 0x1.

cpu0.VINITHI

Reset value of SCTLR.V.
Type: bool. Default value: 0x0.

Parameters for Cluster_ARM_Cortex-A34**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.
Type: bool. Default value: 0x1.

BROADCASTINNER

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

DBGROMADDR

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

Type: `int`. Default value: `0x0`.

DBGROMADDRV

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

Type: `bool`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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.

Type: int. Default value: 0x0.

enable_simulation_performance_optimizations

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).

Type: bool. Default value: 0x1.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: bool. Default value: 0x1.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state_modelled=true.

Type: int. Default value: 0x0.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state_modelled=true.

Type: int. Default value: 0x0.

icache-miss_latency

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.

Type: int. Default value: 0x0.

icache-prefetch_enabled

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

Type: bool. Default value: 0x0.

icache-read_access_latency

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.

Type: int. Default value: 0x0.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

l2cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

l2cache-write_latency

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.

Type: `int`. Default value: `0x0`.

PERIPHBASE

Base address of peripheral memory space.

Type: `int`. Default value: `0x13080000`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.14 ARMCortexA35x1CT

ARMCortexA35x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-100: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA35x1CT contains the following CADI targets:

- ARM_Cortex-A35
- Cluster_ARM_Cortex-A35
- PVCache
- TlbCadi

ARMCortexA35x1CT contains the following MTI components:

- [ARM_Cortex-A35](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

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 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](#).
- 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`.
- 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.

- The following components also exist:
 - ARMCortexA35x2CT.
 - ARMCortexA35x3CT.
 - ARMCortexA35x4CT.

The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-3).

Ports for ARMCortexA35x1CT

Table 4-101: 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 clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	Signal	Slave	Signals the clearing of an external global exclusive monitor
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
commr[4]	Signal	Master	Receive portion of Data Transfer Register full.
commtx[4]	Signal	Master	Transmit portion of Data Transfer Register empty.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cp15sdisable2[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[4]	Signal	Slave	CPU power on reset.

Name	Protocol	Type	Description
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.
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
standbywfi2	Signal	Master	This signal indicated all cores and L2 are idles and in low power state.

Name	Protocol	Type	Description
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-A35

cpu0.AA64nAA32

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

Type: bool. Default value: 0x1.

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CFGTE

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

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE2

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.max_code_cache_mb

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).

Type: int. Default value: 0x100.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: int. Default value: 0xf000000.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: int. Default value: 0x3c.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A35

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTINNER

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

DBGROMADDR

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

Type: `int`. Default value: `0x0`.

DBGROMADDRV

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

Type: `bool`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.
Type: `int`. Default value: `0x8000`.

dcache-snoop_data_transfer_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.
Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.
Type: `int`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).
Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).
Type: `bool`. Default value: `0x1`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

PERIPHBASE

Base address of peripheral memory space.

Type: `int`. Default value: `0x13080000`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

TLB latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

use_architectural_names

Use names SP/LR/PC instead of R13/R14/R15.
Type: `bool`. Default value: `0x0`.

walk_cache_latency

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

4.5.15 ARM Cortex A53x1CT

ARM Cortex A53x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-102: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM Cortex A53x1CT contains the following CADI targets:

- `ARM_Cortex-A53`
- `Cluster_ARM_Cortex-A53`
- `PVCache`
- `TlbCadi`

ARM Cortex A53x1CT contains the following MTI components:

- [ARM_Cortex-A53](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

About ARMCortexA53x1CT

- The following components also exist:
 - ARMCortexA53x2CT.
 - ARMCortexA53x3CT.
 - ARMCortexA53x4CT.

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](#).
- 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 DBGRSTREQ 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 4-103: Ports

Name	Protocol	Type	Description
aa64naa32[4]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port.
broadcastcachemaint	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	Signal	Slave	Enable broadcasting of Inner Shareable transactions.
broadcastouter	Signal	Slave	Enable broadcasting of Outer Shareable transactions.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	Signal	Slave	Signals the clearing of an external global exclusive monitor

Name	Protocol	Type	Description
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
CNTHPIRQ[4]	Signal	Master	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.
dbgllrstdisable	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.
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.

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.
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.
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-A53

cpu0.AA64nAA32

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

Type: bool. Default value: 0x1.

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.
Type: bool. Default value: 0x0.

cpu0.max_code_cache_mb

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).
Type: int. Default value: 0x100.

cpu0.min_sync_level

Force minimum syncLevel (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll).
Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.
Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.
Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.
Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.
Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: int. Default value: 0xf000000.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: int. Default value: 0x3c.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: int. Default value: 0xab.

cpu0.trace_special_hlt_imm16

For this HLT number, IF enable_trace_special_hlt_imm16=true, skip performing usual HLT execution but call MTI trace if registered.
Type: int. Default value: 0xf000.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: bool. Default value: 0x0.

cpu0.vfp-present

Set whether the model has VFP support.
Type: bool. Default value: 0x1.

cpu0.VINITHI

Reset value of SCTLR.V.
Type: bool. Default value: 0x0.

Parameters for Cluster_ARM_Cortex-A53**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.
Type: bool. Default value: 0x1.

BROADCASTINNER

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

DBGROMADDR

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

Type: `int`. Default value: `0x0`.

DBGROMADDRV

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

Type: `bool`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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.

Type: int. Default value: 0x0.

enable_simulation_performance_optimizations

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).

Type: bool. Default value: 0x1.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: bool. Default value: 0x1.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state_modelled=true.

Type: int. Default value: 0x0.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state_modelled=true.

Type: int. Default value: 0x0.

icache-miss_latency

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.

Type: int. Default value: 0x0.

icache-prefetch_enabled

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

Type: bool. Default value: 0x0.

icache-read_access_latency

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.

Type: int. Default value: 0x0.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

patch_level

Cosmetic change to Revision field in MIDR/MIDR_EL1. Corresponds to the patch number Y in rXpY.

Type: `int`. Default value: `0x1`.

PERIPHBASE

Base address of peripheral memory space.

Type: `int`. Default value: `0x13080000`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

revision_number

Cosmetic change to Variant field in MIDR/MIDR_EL1. Corresponds to the revision number X in rXpY.

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.16 ARM Cortex A55CT

ARM Cortex A55CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-104: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM Cortex A55CT contains the following CADI targets:

- ARM_Cortex-A55
- Cluster_ARM_Cortex-A55
- PVCache
- TlbCadi

ARM Cortex A55CT contains the following MTI components:

- [ARM_Cortex-A55](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `CMO_broadcast_when_cache_state_modelling_disabled`
- `dcache-prefetch_enabled`
- `enable_simulation_performance_optimizations`
- `has_dot_product`
- `pchannel_treat_simreset_as_poreset`
- `use_architectural_names`
- `walk_cache_latency`

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 DBGPWRDUP and DBGIRSTREQ are not implemented, but DBGPWRUPREQ 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 following models also exist, with a fixed number of cores per cluster:

- ARMCortexA55x1CT.
- ARMCortexA55x2CT.
- ARMCortexA55x3CT.
- ARMCortexA55x4CT.
- ARMCortexA55x8CT.

The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-7).

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 ARMCortexA55CT

Table 4-105: Ports

Name	Protocol	Type	Description
aa64naa32[8]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port.
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[8]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[8]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[8]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[8]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[8]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[8]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[8]	Signal	Master	Timer signals to SOC.
commirq[8]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[8]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable[8]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[8]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[8]	Signal	Slave	Disable cryptography extensions after reset.

Name	Protocol	Type	Description
cti[8]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[8]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[8]	Signal	Master	No power-down request.
dbgpwrupreq[8]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[8]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[8]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[8]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[8]	PChannel	Slave	PChannels for cores
pmuirq[8]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[8]	Signal	Slave	Per core RAM Error Interrupt.
reset[8]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[8]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[8]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[8]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.

Name	Protocol	Type	Description
vcpumntirq[8]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[8]	Signal	Slave	Virtualised FIQ.
vinithi[8]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[8]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[8]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-A55

cpu0.AA64nAA32

Register width configuration at reset. 0, AArch32. 1, AArch64.
Type: bool. Default value: 0x1.

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.
Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.
Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.
Type: int. Default value: 0x0.

cpu0.l2cache-miss_latency

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.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-size`

L2 Cache size in bytes.

Type: `int`. Default value: `0x40000`.

`cpu0.l2cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-snoop_issue_latency`

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Value of RVBAR_ELx register.
Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.
Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.
Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.
Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: int. Default value: 0xf000000.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: int. Default value: 0x3c.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: int. Default value: 0xab.

cpu0.trace_special_hlt_imm16

For this HLT number, IF enable_trace_special_hlt_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: int. Default value: 0xf000.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool. Default value: 0x0.

cpu0.vfp-present

Set whether the model has VFP support.

Type: bool. Default value: 0x1.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: bool. Default value: 0x0.

Parameters for Cluster_ARM_Cortex-A55**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: bool. Default value: 0x1.

BROADCASTCACHEMAINT

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

Type: bool. Default value: 0x1.

BROADCASTOUTER

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

Type: bool. Default value: 0x1.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: bool. Default value: 0x1.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: int. Default value: 0x0.

CLUSTER_ID

Processor cluster ID value.

Type: int. Default value: 0x0.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

`cpi_div`

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

`cpi_mul`

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

`dcache-hit_latency`

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`dcache-maintenance_latency`

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`dcache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-prefetch_enabled`

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

Type: `bool`. Default value: `0x0`.

`dcache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-size`

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

force_zero_PSTATE_PAN

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`.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.17 ARMCortexA55CT_CortexA75CT

ARMCortexA55CT_CortexA75CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-106: IP revisions support

Revision	Quality level
CortexA55 r1p0	Full support
CortexA75 r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

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

ARMCortexA55CT_CortexA75CT contains the following MTI components:

- [ARM_Cortex-A55](#)
- [ARM_Cortex-A75](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `subcluster0.use_architectural_names`
- `subcluster1.use_architectural_names`

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:

- [4.5.16 ARMCortexA55CT](#) on page 389.
- [4.5.26 ARMCortexA75CT](#) on page 532.

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 ARMCortexA55CT_CortexA75CT

Table 4-107: 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 is 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.

Name	Protocol	Type	Description
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
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.

Name	Protocol	Type	Description
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 ARM_Cortex-A55

cpu0.AA64nAA32

Register width configuration at reset. 0, AArch32. 1, AArch64.
Type: bool. Default value: 0x1.

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.
Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.
Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.
Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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.
Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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.
Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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.
Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.
Type: `int`. Default value: `0x40000`.

cpu0.l2cache-snoop_data_transfer_latency

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.
Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.
Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.RVBARADDR`

Value of `RVBAR_ELx` register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: `int`. Default value: `0xf0000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.
Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.
Type: `bool`. Default value: `0x1`.

cpu0.VINITHI

Reset value of SCTLR.V.
Type: `bool`. Default value: `0x0`.

Parameters for ARM_Cortex-A75**cpu0.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.
Type: `bool`. Default value: `0x1`.

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: `bool`. Default value: `0x0`.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: `bool`. Default value: `0x0`.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.
Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-write_latency

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.

Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).

Type: int. Default value: 0x100.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.
Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: `bool`. Default value: `0x0`.

cpu0.VINITHI

Reset value of SCTLR.V.
Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A55_Cortex-A75**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.
Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAIN

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.
Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.
Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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.

Type: `int`. Default value: `0x0`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

Parameters for Subcluster_ARM_Cortex-A55**subcluster0.cpi_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster0.cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster0.dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-miss_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

subcluster0.dcache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-read_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

subcluster0.dcache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-write_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

subcluster0.has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

subcluster0.icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster0.icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster0.icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster0.icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

subcluster0.icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster0.icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster0.icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

subcluster0.NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

subcluster0.ptw_latency

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

Type: `int`. Default value: `0x0`.

subcluster0.tlb_latency

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

Type: `int`. Default value: `0x0`.

subcluster0.use_architectural_names

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

Type: `bool`. Default value: `0x0`.

subcluster0.walk_cache_latency

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

Type: `int`. Default value: `0x0`.

Parameters for Subcluster_ARM_Cortex-A75

subcluster1.CCSIDR-L1D_override

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

Type: `int`. Default value: `0x0`.

subcluster1.CCSIDR-L1I_override

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

Type: `int`. Default value: `0x0`.

subcluster1.CCSIDR-L2_override

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

Type: `int`. Default value: `0x0`.

subcluster1.cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster1.cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster1.dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-prefetch_enabled

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

Type: bool. Default value: 0x0.

subcluster1.dcache-read_access_latency

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.

Type: int. Default value: 0x0.

subcluster1.dcache-read_latency

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.

Type: int. Default value: 0x0.

subcluster1.dcache-snoop_data_transfer_latency

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.

Type: int. Default value: 0x0.

subcluster1.dcache-write_access_latency

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.

Type: int. Default value: 0x0.

subcluster1.dcache-write_latency

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.

Type: int. Default value: 0x0.

subcluster1.force_zero_PSTATE_PAN

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.

Type: bool. Default value: 0x0.

subcluster1.icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state_modelled=true.

Type: int. Default value: 0x0.

subcluster1.icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state_modelled=true.

Type: `int`. Default value: `0x0`.

subcluster1.icache-miss_latency

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.

Type: `int`. Default value: `0x0`.

subcluster1.icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

subcluster1.icache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

subcluster1.icache-read_latency

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.

Type: `int`. Default value: `0x0`.

subcluster1.NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

subcluster1.ptw_latency

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

Type: `int`. Default value: `0x0`.

subcluster1.tlb_latency

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

Type: `int`. Default value: `0x0`.

subcluster1.use_architectural_names

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

Type: `bool`. Default value: `0x0`.

subcluster1.walk_cache_latency

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

Type: `int`. Default value: `0x0`.

Related information[Arm DynamIQ Shared Unit Technical Reference Manual](#)**4.5.18 ARM CortexA55CT_CortexA76CT**

ARM CortexA55CT_CortexA76CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-108: IP revisions support

Revision	Quality level
CortexA55 r1p0	Full support
CortexA76 r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM CortexA55CT_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

ARM CortexA55CT_CortexA76CT contains the following MTI components:

- [ARM_Cortex-A55](#)
- [ARM_Cortex-A76](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `subcluster0.use_architectural_names`
- `subcluster1.use_architectural_names`

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:

- [4.5.16 ARMCortexA55CT](#) on page 389.
- [4.5.28 ARMCortexA76CT](#) on page 558.

Ports for ARMCortexA55CT_CortexA76CT

Table 4-109: 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	DynamiQ port to obtain end address of valid peripheral address range (exclusive).
<code>ASTARTMP</code>	Value_64	Slave	DynamiQ 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.

Name	Protocol	Type	Description
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[11]	Signal	Slave	This signal is 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.
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.

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[11]	PChannel	Slave	PChannels for cores
pmbirq[11]	Signal	Master	Interrupt signal from statistical profiling unit.
pmuirq[11]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[11]	Signal	Slave	Per core RAM Error Interrupt.
reset[11]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[11]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[11]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynaMIQ.
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 ARM_Cortex-A55

cpu0.AA64nAA32

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

Type: `bool`. Default value: `0x1`.

cpu0.CFGEND

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

Type: `bool`. Default value: `0x0`.

cpu0.CFGTE

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

Type: `bool`. Default value: `0x0`.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x40000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Value of `RVBAR_ELx` register.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

`cpu0.vfp-present`

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

`cpu0.VINITHI`

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for ARM_Cortex-A76

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: `bool`. Default value: `0x0`.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.
Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.
Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.
Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.
Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.
Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

`cpu0.VINITHI`

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A55_Cortex-A76

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive (corresponds to `AENDMP` input signal).

Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive (corresponds to `ASTARTMP` input signal).

Type: `int`. Default value: `0x0`.

Parameters for Subcluster_ARM_Cortex-A55**subcluster0.cpi_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster0.cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster0.dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-miss_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

subcluster0.dcache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-read_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

subcluster0.dcache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-write_latency

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.

Type: int. Default value: 0x0.

subcluster0.force_zero_PSTATE_PAN

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.

Type: bool. Default value: 0x0.

subcluster0.has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: int. Default value: 0x2.

subcluster0.icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state_modelled=true.

Type: int. Default value: 0x0.

subcluster0.icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state_modelled=true.

Type: int. Default value: 0x0.

subcluster0.icache-miss_latency

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.

Type: int. Default value: 0x0.

subcluster0.icache-prefetch_enabled

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

Type: bool. Default value: 0x0.

subcluster0.icache-read_access_latency

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.

Type: int. Default value: 0x0.

subcluster0.icache-read_latency

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.

Type: int. Default value: 0x0.

subcluster0.icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

subcluster0.NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

subcluster0.ptw_latency

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

Type: `int`. Default value: `0x0`.

subcluster0.tlb_latency

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

Type: `int`. Default value: `0x0`.

subcluster0.use_architectural_names

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

Type: `bool`. Default value: `0x0`.

subcluster0.walk_cache_latency

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

Type: `int`. Default value: `0x0`.

Parameters for Subcluster_ARM_Cortex-A76**subcluster1.cpi_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster1.cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster1.dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

subcluster1.dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`. Default value: `0x0`.

subcluster1.ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`. Default value: `0x0`.

subcluster1.ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`. Default value: `0x0`.

subcluster1.ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`. Default value: `0x0`.

subcluster1.force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

subcluster1.has_statistical_profiling

Whether Statistical Based Profiling is implemented (FEAT_SPE).

Type: `bool`. Default value: `0x1`.

subcluster1.icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster1.icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster1.icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

subcluster1.icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

subcluster1.ptw_latency

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

Type: `int`. Default value: `0x0`.

subcluster1.tlb_latency

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

Type: `int`. Default value: `0x0`.

subcluster1.treat-dcache-cmos-to-pou-as-nop

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.

Type: `int`. Default value: `0x0`.

subcluster1.use_architectural_names

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

Type: `bool`. Default value: `0x0`.

subcluster1.walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.19 ARM CortexA55CT_CortexA78CT

ARM CortexA55CT_CortexA78CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-110: IP revisions support

Revision	Quality level
CortexA55 r1p0	Full support
CortexA78 r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM CortexA55CT_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

ARM CortexA55CT_CortexA78CT contains the following MTI components:

- [ARM_Cortex-A55](#)
- [ARM_Cortex-A78](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)

- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `subcluster0.use_architectural_names`
- `subcluster1.use_architectural_names`

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:

- [4.5.16 ARMCortexA55CT](#) on page 389.
- [4.5.32 ARMCortexA78CT](#) on page 609.

Ports for ARMCortexA55CT_CortexA78CT

Table 4-111: Ports

Name	Protocol	Type	Description
aa64naa32[11]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port
AENDMP	Value_64	Slave	DynamlQ port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	DynamlQ port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[11]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[11]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:14] of the MPIDR.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[11]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[11]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[11]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[11]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[11]	Signal	Master	Timer signals to SOC.
commirq[11]	Signal	Master	Interrupt signal from debug communications channel.
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.
dbgppwrupreq[11]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[11]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[11]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[11]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[11]	PChannel	Slave	PChannels for cores
pmbirq[11]	Signal	Master	Interrupt signal from statistical profiling unit.
pmuirq[11]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[11]	Signal	Slave	Per core RAM Error Interrupt.
reset[11]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[11]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[11]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
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.

Name	Protocol	Type	Description
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 ARM_Cortex-A55

cpu0.AA64nAA32

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

Type: bool. Default value: 0x1.

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CFGTE

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

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-miss_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x40000`.

cpu0.l2cache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool. Default value: 0x0.

cpu0.vfp-present

Set whether the model has VFP support.

Type: bool. Default value: 0x1.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: bool. Default value: 0x0.

Parameters for ARM_Cortex-A78**cpu0.CFGEND**

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

Type: bool. Default value: 0x0.

cpu0.CFGTE

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-miss_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_latency`

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`.
Type: `int`. Default value: `0x0`.

`cpu0.l2cache-size`

L2 Cache size in bytes.
Type: `int`. Default value: `0x80000`.

`cpu0.l2cache-snoop_data_transfer_latency`

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`.
Type: `int`. Default value: `0x0`.

`cpu0.l2cache-snoop_issue_latency`

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_access_latency`

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`.
Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_latency`

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`.
Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).
Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

Force minimum syncLevel (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll).
Type: `int`. Default value: `0x0`.

`cpu0.RVBARADDR`

Value of RVBAR_ELx register.
Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.
Type: `int`. Default value: `0xf000`.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.
Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.
Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.
Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: int. Default value: 0xf000000.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: int. Default value: 0x3c.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: int. Default value: 0xab.

cpu0.trace_special_hlt_imm16

For this HLT number, if enable_trace_special_hlt_imm16=true, skip performing usual HLT execution but call MTI trace if registered.
Type: int. Default value: 0xf000.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: bool. Default value: 0x0.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A55_Cortex-A78**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamlQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.
Type: `int`. Default value: `0x80000`.

l3cache-snoop_data_transfer_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.
Type: `int`. Default value: `0x0`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.
Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive (corresponds to `AENDMP` input signal).
Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive (corresponds to `ASTARTMP` input signal).
Type: `int`. Default value: `0x0`.

Parameters for Subcluster_ARM_Cortex-A55**subcluster0.cpi_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster0.cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster0.dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

subcluster0.dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

subcluster0.dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-write_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

subcluster0.has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

subcluster0.icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state_modelled=true.

Type: `int`. Default value: `0x0`.

subcluster0.icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state_modelled=true.

Type: `int`. Default value: `0x0`.

subcluster0.icache-miss_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

subcluster0.icache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.icache-read_latency

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`.
Type: `int`. Default value: `0x0`.

subcluster0.icache-size

L1 I-Cache size in bytes.
Type: `int`. Default value: `0x8000`.

subcluster0.NUM_CORES

Number of cores per cluster.
Type: `int`. Default value: `0x1`.

subcluster0.ptw_latency

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

subcluster0.tlb_latency

TLB latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

subcluster0.use_architectural_names

Use names SP/LR/PC instead of R13/R14/R15.
Type: `bool`. Default value: `0x0`.

subcluster0.walk_cache_latency

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

Parameters for Subcluster_ARM_Cortex-A78

subcluster1.cpi_div

Divider for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

subcluster1.cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

subcluster1.dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

subcluster1.dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

subcluster1.dcache-miss_latency

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.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

subcluster1.dcache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-read_latency

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.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

subcluster1.dcache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-write_latency

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.

Type: `int`. Default value: `0x0`.

subcluster1.ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`. Default value: `0x0`.

subcluster1.ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`. Default value: `0x0`.

subcluster1.ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`. Default value: `0x0`.

subcluster1.ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`. Default value: `0x0`.

subcluster1.force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

subcluster1.has_coherent_icache

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`. Default value: `0x0`.

subcluster1.has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

subcluster1.icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster1.icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster1.icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

subcluster1.icache-read_access_latency

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`.
Type: `int`. Default value: `0x0`.

`subcluster1.icache-read_latency`

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`.
Type: `int`. Default value: `0x0`.

`subcluster1.icache-size`

L1 I-Cache size in bytes.
Type: `int`. Default value: `0x10000`.

`subcluster1.NUM_CORES`

Number of cores per cluster.
Type: `int`. Default value: `0x1`.

`subcluster1.ptw_latency`

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

`subcluster1.tlb_latency`

TLB latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

`subcluster1.treat-dcache-cmos-to-pou-as-nop`

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.
Type: `int`. Default value: `0x0`.

`subcluster1.use_architectural_names`

Use names SP/LR/PC instead of R13/R14/R15.
Type: `bool`. Default value: `0x0`.

`subcluster1.walk_cache_latency`

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

4.5.20 ARM Cortex A57x1CT

ARM Cortex A57x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-112: IP revisions support

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA57x1CT contains the following CADI targets:

- ARM_Cortex-A57
- Cluster_ARM_Cortex-A57
- PVCache
- TlbCadi

ARMCortexA57x1CT contains the following MTI components:

- [ARM_Cortex-A57](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

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 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](#).

- 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 `event_i` and `event_o` 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 DBGPWDUP, DBGWRUPREQ, DBGNOPWRDWN, and DBGRSTREQ debug power management signals.
- ECC and parity schemes are hardware-specific so are not supported.

Ports for ARMCortexA57x1CT

Table 4-113: Ports

Name	Protocol	Type	Description
aa64naa32[4]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port.
broadcastcachemaint	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	Signal	Slave	Enable broadcasting of Inner Shareable transactions.
broadcastouter	Signal	Slave	Enable broadcasting of Outer Shareable transactions.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	Signal	Slave	Signals the clearing of an external global exclusive monitor
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
CNTHPIRQ[4]	Signal	Master	The per-EL counter signal.
CNTPNSIRQ[4]	Signal	Master	The per-EL counter signal.
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.

Name	Protocol	Type	Description
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgack[4]	Signal	Master	External debug interface.
dbgen[4]	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	These signals relate to core power down.
dbgpwrupreq[4]	Signal	Master	These signals relate to core power down.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
l2flushdone	Signal	Master	Flush of L2 memory system complete
l2flushreq	Signal	Slave	Request flush of L2 memory system.
l2reset	Signal	Slave	Reset the shared L2 memory system controller.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
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.

Name	Protocol	Type	Description
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
standbywfil2	Signal	Master	Indicate that all the individual processors and the L2 systems are in a WFI state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtual FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtual IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Processor Virtual System Error Interrupt request.

Parameters for ARM_Cortex-A57

cpu0.AA64nAA32

Register width configuration at reset. 0, AArch32. 1, AArch64.
Type: bool. Default value: 0x1.

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.
Type: bool. Default value: 0x0.

cpu0.max_code_cache_mb

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).
Type: int. Default value: 0x100.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: int. Default value: 0xf000000.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: int. Default value: 0x3c.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A57

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTINNER

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

DBGROMADDR

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

Type: `int`. Default value: `0x22000000`.

DBGROMADDRV

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

Type: `bool`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: bool. Default value: 0x0.

icache-read_access_latency

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.

Type: int. Default value: 0x0.

icache-read_latency

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.

Type: int. Default value: 0x0.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: bool. Default value: 0x0.

l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

l2cache-miss_latency

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.

Type: int. Default value: 0x0.

l2cache-read_access_latency

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.

Type: int. Default value: 0x0.

l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

PERIPHBASE

Base address of peripheral memory space.

Type: `int`. Default value: `0x13080000`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.21 ARM Cortex A65 AECT

ARM Cortex A65 AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-114: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM Cortex A65 AECT contains the following CADI targets:

- ARM_Cortex-A65AE
- Cluster_ARM_Cortex-A65AE
- PVCache
- TlbCadi

ARM Cortex A65 AECT contains the following MTI components:

- [ARM_Cortex-A65AE](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

About ARM Cortex A65 AECT

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

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. The following models also exist, with a fixed number of cores per cluster:

- `ARMCortexA65AECTx2CT`.
- `ARMCortexA65AECTx4CT`.
- `ARMCortexA65AECTx6CT`.
- `ARMCortexA65AECTx8CT`.

The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-7).

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](#).

AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following caveats:

- 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 4-115: 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 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.
<code>core_clk_in[8]</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[8]</code>	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.

Name	Protocol	Type	Description
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.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.

Name	Protocol	Type	Description
ticks[16]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[16]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[16]	Signal	Slave	Virtualised FIQ.
vinithi[16]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[16]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[16]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-A65AE

cpu0.thread0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.thread0.enable_single_thread_at_reset

Enable single thread after reset and keep other thread in reset.

Type: bool. Default value: 0x0.

cpu0.thread0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.thread0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.thread0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.thread0.l2cache-miss_latency

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.

Type: int. Default value: 0x0.

cpu0.thread0.l2cache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

`cpu0.thread0.l2cache-read_latency`

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`.
Type: `int`. Default value: `0x0`.

`cpu0.thread0.l2cache-size`

L2 Cache size in bytes.
Type: `int`. Default value: `0x40000`.

`cpu0.thread0.l2cache-snoop_data_transfer_latency`

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`.
Type: `int`. Default value: `0x0`.

`cpu0.thread0.l2cache-snoop_issue_latency`

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

`cpu0.thread0.l2cache-write_access_latency`

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`.
Type: `int`. Default value: `0x0`.

`cpu0.thread0.l2cache-write_latency`

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`.
Type: `int`. Default value: `0x0`.

`cpu0.thread0.max_code_cache_mb`

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).
Type: `int`. Default value: `0x100`.

`cpu0.thread0.min_sync_level`

Force minimum `syncLevel` (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll).
Type: `int`. Default value: `0x0`.

`cpu0.thread0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.
Type: `int`. Default value: `0x000`.

`cpu0.thread0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.
Type: `int`. Default value: `0x000`.

cpu0.thread0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.
Type: `int`. Default value: `0x123456`.

cpu0.thread0.semihosting-cmd_line

Command line available to semihosting calls.
Type: `string`. Default value: `""`.

cpu0.thread0.semihosting-cwd

Base directory for semihosting file access.
Type: `string`. Default value: `""`.

cpu0.thread0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: `bool`. Default value: `0x1`.

cpu0.thread0.semihosting-heap_base

Virtual address of heap base.
Type: `int`. Default value: `0x0`.

cpu0.thread0.semihosting-heap_limit

Virtual address of top of heap.
Type: `int`. Default value: `0xf000000`.

cpu0.thread0.semihosting-stack_base

Virtual address of base of descending stack.
Type: `int`. Default value: `0x10000000`.

cpu0.thread0.semihosting-stack_limit

Virtual address of stack limit.
Type: `int`. Default value: `0xf000000`.

cpu0.thread0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: `int`. Default value: `0x3c`.

cpu0.thread0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: `int`. Default value: `0xab`.

cpu0.thread0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.
Type: `int`. Default value: `0xf000`.

cpu0.thread0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: `bool`. Default value: `0x0`.

cpu0.thread0.vfp-present

Set whether the model has VFP support.
Type: `bool`. Default value: `0x1`.

Parameters for Cluster_ARM_Cortex-A65AE

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: bool. Default value: 0x1.

BROADCASTCACHEMAINT

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

Type: bool. Default value: 0x1.

BROADCASTOUTER

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

Type: bool. Default value: 0x1.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: bool. Default value: 0x1.

CLUSTER_ID

Processor cluster ID value.

Type: int. Default value: 0x0.

cluster_patch_level

Cosmetic change to patch number in CLUSTERIDR. Corresponds to the Y in rXpY.

Type: int. Default value: 0x0.

cluster_revision_number

Cosmetic change to revision number in CLUSTERIDR, Corresponds to the X in rXpY.

Type: int. Default value: 0x0.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: int. Default value: 0x1.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: int. Default value: 0x1.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: int. Default value: 0x1.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_lock_step

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

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x400000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

NUM_CORES

Number of cores per cluster.
Type: `int`. Default value: `0x2`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.
Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).
Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).
Type: `int`. Default value: `0x0`.

ptw_latency

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

tlb_latency

TLB latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

use_architectural_names

Use names SP/LR/PC instead of R13/R14/R15.
Type: `bool`. Default value: `0x0`.

walk_cache_latency

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

4.5.22 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 4-116: IP revisions support

Revision	Quality level
CortexA65AE r0p0	Full support
CortexA76AE r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

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

ARMCortexA65AECT_CortexA76AECT contains the following MTI components:

- [ARM_Cortex-A65AE](#)
- [ARM_Cortex-A76AE](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `subcluster0.use_architectural_names`
- `subcluster1.use_architectural_names`

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 threads in `subcluster0`. For example:
 - `<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:

- [4.5.21 ARMCortexA65AECT](#) on page 468.
- [4.5.27 ARMCortexA76AECT](#) on page 545.

Ports for ARMCortexA65AECT_CortexA76AECT

Table 4-117: Ports

Name	Protocol	Type	Description
<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[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.

Name	Protocol	Type	Description
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.
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[10]	PChannel	Slave	PChannels for cores
pmbirq[16]	Signal	Master	Interrupt signal from statistical profiling unit.
pmuirq[16]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[16]	Signal	Slave	Per core RAM Error Interrupt.
reset[10]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[16]	Value_64	Slave	Reset vector base address.

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[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_Cortex-A65AE

cpu0.thread0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.thread0.enable_single_thread_at_reset

Enable single thread after reset and keep other thread in reset.
Type: bool. Default value: 0x0.

cpu0.thread0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.
Type: bool. Default value: 0x0.

cpu0.thread0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.
Type: int. Default value: 0x0.

cpu0.thread0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.
Type: int. Default value: 0x0.

cpu0.thread0.l2cache-miss_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x40000`.

cpu0.thread0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.thread0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.thread0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.thread0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.thread0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.thread0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.thread0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.thread0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.thread0.semihosting-heap_base

Virtual address of heap base.

Type: int. Default value: 0x0.

cpu0.thread0.semihosting-heap_limit

Virtual address of top of heap.

Type: int. Default value: 0xf000000.

cpu0.thread0.semihosting-stack_base

Virtual address of base of descending stack.

Type: int. Default value: 0x10000000.

cpu0.thread0.semihosting-stack_limit

Virtual address of stack limit.

Type: int. Default value: 0xf000000.

cpu0.thread0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: int. Default value: 0x3c.

cpu0.thread0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: int. Default value: 0xab.

cpu0.thread0.trace_special_hlt_imm16

For this HLT number, IF enable_trace_special_hlt_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.thread0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.thread0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for ARM_Cortex-A76AE**cpu0.CFGEND**

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

Type: `bool`. Default value: `0x0`.

cpu0.CFGTE

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

Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-size`

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

`cpu0.l2cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-snoop_issue_latency`

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.RVBARADDR`

Value of `RVBAR_ELx` register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

`cpu0.VINITHI`

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A65AE_Cortex-A76AE

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamlQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_lock_step

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

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

`l3cache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-size`

L3 Cache size in bytes.

Type: `int`. Default value: `0x100000`.

`l3cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-snoop_issue_latency`

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`l3cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`pchannel_treat_simreset_as_poreset`

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

`periph_address_end`

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

`periph_address_start`

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

Parameters for Subcluster_ARM_Cortex-A65AE

subcluster0.cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster0.cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster0.dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

subcluster0.dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

subcluster0.dcache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.dcache-write_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

subcluster0.has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

subcluster0.icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state_modelled=true.

Type: `int`. Default value: `0x0`.

subcluster0.icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state_modelled=true.

Type: `int`. Default value: `0x0`.

subcluster0.icache-miss_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

subcluster0.icache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.icache-read_latency

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.

Type: `int`. Default value: `0x0`.

subcluster0.icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

subcluster0.NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x2`.

subcluster0.ptw_latency

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

Type: `int`. Default value: `0x0`.

subcluster0.tlb_latency

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

Type: `int`. Default value: `0x0`.

subcluster0.use_architectural_names

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

Type: `bool`. Default value: `0x0`.

subcluster0.walk_cache_latency

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

Type: `int`. Default value: `0x0`.

Parameters for Subcluster_ARM_Cortex-A76AE**subcluster1.cpi_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster1.cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

subcluster1.dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

subcluster1.dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: `boo1`. Default value: `0x0`.

subcluster1.ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: `boo1`. Default value: `0x0`.

subcluster1.ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `boo1`. Default value: `0x0`.

subcluster1.ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `boo1`. Default value: `0x0`.

subcluster1.force_zero_PSTATE_PAN

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.

Type: `boo1`. Default value: `0x0`.

subcluster1.has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

subcluster1.has_statistical_profiling

Whether Statistical Based Profiling is implemented (FEAT_SPE).

Type: `boo1`. Default value: `0x1`.

subcluster1.icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster1.icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

subcluster1.icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

subcluster1.icache-prefetch_enabled

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

Type: `boo1`. Default value: `0x0`.

subcluster1.icache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

subcluster1.icache-read_latency

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.

Type: `int`. Default value: `0x0`.

subcluster1.NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x2`.

subcluster1.ptw_latency

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

Type: `int`. Default value: `0x0`.

subcluster1.tlb_latency

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

Type: `int`. Default value: `0x0`.

subcluster1.treat-dcache-cmos-to-pou-as-nop

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.

Type: `int`. Default value: `0x0`.

subcluster1.use_architectural_names

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

Type: `bool`. Default value: `0x0`.

subcluster1.walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.23 ARMCortexA65CT

ARMCortexA65CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-118: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA65CT contains the following CADI targets:

- ARM_Cortex-A65
- Cluster_ARM_Cortex-A65
- PVCache
- TlbCadi

ARMCortexA65CT contains the following MTI components:

- [ARM_Cortex-A65](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

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 DBGWRDUP and DBGRSTREQ are not implemented, but DBGWRUPREQ 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 component has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. The following components also exist, with a fixed number of cores per cluster:

- ARMCortexA65x1CT.
- ARMCortexA65x2CT.
- ARMCortexA65x3CT.
- ARMCortexA65x4CT.

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 ARMCortexA65CT

Table 4-119: 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.

Name	Protocol	Type	Description
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[16]	Signal	Slave	This signal is for EE bit initialisation.
cfgte[16]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
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.
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.

Name	Protocol	Type	Description
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[8]	PChannel	Slave	PChannels for cores
pmuirq[16]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[16]	Signal	Slave	Per core RAM Error Interrupt.
reset[8]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[16]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[16]	Signal	Slave	Per core virtual System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[16]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[16]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[16]	Signal	Slave	Virtualised FIQ.
vinithi[16]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[16]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[16]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-A65

`cpu0.thread0.CRYPTODISABLE`

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.thread0.enable_single_thread_at_reset

Enable single thread after reset and keep other thread in reset.

Type: `bool`. Default value: `0x0`.

cpu0.thread0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.thread0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x40000`.

cpu0.thread0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-write_access_latency

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.

Type: int. Default value: 0x0.

cpu0.thread0.l2cache-write_latency

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.

Type: int. Default value: 0x0.

cpu0.thread0.max_code_cache_mb

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).

Type: int. Default value: 0x100.

cpu0.thread0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.thread0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.thread0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.thread0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.thread0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.thread0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.thread0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.thread0.semihosting-heap_base

Virtual address of heap base.

Type: int. Default value: 0x0.

cpu0.thread0.semihosting-heap_limit

Virtual address of top of heap.

Type: int. Default value: 0xf000000.

cpu0.thread0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.thread0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf0000000`.

cpu0.thread0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.thread0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.thread0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.thread0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.thread0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for Cluster_ARM_Cortex-A65

BROADCASTATOMIC

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

cluster_patch_level

Cosmetic change to patch number in CLUSTERIDR. Corresponds to the Y in rXpY.

Type: `int`. Default value: `0x0`.

cluster_revision_number

Cosmetic change to revision number in CLUSTERIDR, Corresponds to the X in rXpY.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

dcache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x400000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.24 ARM Cortex A72x1CT

ARM Cortex A72x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-120: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM Cortex A72x1CT contains the following CADI targets:

- `ARM_Cortex-A72`
- `Cluster_ARM_Cortex-A72`
- `PVCache`
- `TlbCadi`

ARM Cortex A72x1CT contains the following MTI components:

- [ARM_Cortex-A72](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

About ARM Cortex A72x1CT

- 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 4-121: 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 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	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.

Name	Protocol	Type	Description
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
commrx[4]	Signal	Master	Receive portion of Data Transfer Register full.
commtx[4]	Signal	Master	Transmit portion of Data Transfer Register empty.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[4]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgack[4]	Signal	Master	External debug interface.
dbgen[4]	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	These signals relate to core power down.
dbgpwrupreq[4]	Signal	Master	These signals relate to core power down.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
l2flushdone	Signal	Master	Flush of L2 memory system complete.
l2flushreq	Signal	Slave	Request flush of L2 memory system.
l2reset	Signal	Slave	Reset the shared L2 memory system controller.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
rei[4]	Signal	Slave	Individual processor RAM Error Interrupt signal input.

Name	Protocol	Type	Description
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.
standbywfi2	Signal	Master	Indicate that all the individual processors and the L2 systems are in a WFI state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtual FIQ
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtual IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Processor Virtual System Error Interrupt request.

Parameters for ARM_Cortex-A72

cpu0.AA64nAA32

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

Type: bool. Default value: 0x1.

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CFGTE

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

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.max_code_cache_mb

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).

Type: int. Default value: 0x100.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf0000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A72**BROADCASTCACHEMAINT**

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

Type: `bool`. Default value: `0x1`.

BROADCASTINNER

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

DBGROMADDR

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

Type: `int`. Default value: `0x22000000`.

DBGROMADDRV

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

Type: `bool`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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.

Type: `int`. Default value: `0x0`.

dcache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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.

Type: `int`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

PERIPHBASE

Base address of peripheral memory space.

Type: `int`. Default value: `0x13080000`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.25 ARM Cortex A73x1CT

ARM Cortex A73x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-122: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM Cortex A73x1CT contains the following CADI targets:

- `ARM_Cortex-A73`
- `Cluster_ARM_Cortex-A73`
- `PVCache`
- `TlbCadi`

ARM Cortex A73x1CT contains the following MTI components:

- [ARM_Cortex-A73](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)

- [TLB](#)

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 4-123: 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

Name	Protocol	Type	Description
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
commrx[4]	Signal	Master	Receive portion of Data Transfer Register full.
commtx[4]	Signal	Master	Transmit portion of Data Transfer Register empty.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[4]	Signal	Slave	CPU power on reset.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross trigger matrix port.
dbgack[4]	Signal	Master	External debug interface.
dbgen[4]	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Processor powerup request.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
l2flushdone	Signal	Master	Flush of L2 memory system complete.
l2flushreq	Signal	Slave	Request flush of L2 memory system.
l2reset	Signal	Slave	Level2 reset.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Debug reset.

Name	Protocol	Type	Description
pvbush_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 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 ARM_Cortex-A73

cpu0.AA64nAA32

Register width configuration at reset. 0, AArch32. 1, AArch64.
Type: bool. Default value: 0x1.

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: `bool`. Default value: `0x0`.

`cpu0.CRYPTODISABLE`

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

`cpu0.enable_trace_special_hlt_imm16`

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.RVBARADDR`

Value of `RVBAR_ELx` register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf0000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf0000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.SMPEN

Enable broadcast messages necessary for correct SMP operation at reset.

Type: `bool`. Default value: `0x0`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A73

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTINNER

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

CCSIDR-L1D_override

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

Type: `int`. Default value: `0x0`.

CCSIDR-L1I_override

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

Type: `int`. Default value: `0x0`.

CCSIDR-L2_override

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

DBGROMADDR

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

Type: `int`. Default value: `0x0`.

DBGROMADDRV

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

Type: `bool`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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.

Type: int. Default value: 0x0.

enable_simulation_performance_optimizations

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).

Type: bool. Default value: 0x1.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: bool. Default value: 0x1.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state_modelled=true.

Type: int. Default value: 0x0.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state_modelled=true.

Type: int. Default value: 0x0.

icache-miss_latency

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.

Type: int. Default value: 0x0.

icache-prefetch_enabled

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

Type: bool. Default value: 0x0.

icache-read_access_latency

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.

Type: int. Default value: 0x0.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l2cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

l2cache-write_latency

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.

Type: `int`. Default value: `0x0`.

MIDR

Value of MIDR_EL1 register.

Type: `int`. Default value: `0x411fd090`.

PERIPHBASE

Base address of peripheral memory space.

Type: `int`. Default value: `0x13080000`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

revision_number

Cosmetic change to Variant field in MIDR/MIDR_EL1. Corresponds to the revision number X in rXpY.

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.26 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 4-124: IP revisions support

Revision	Quality level
r3p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA75CT contains the following CADI targets:

- ARM_Cortex-A75
- Cluster_ARM_Cortex-A75
- PVCache
- TlbCadi

ARMCortexA75CT contains the following MTI components:

- [ARM_Cortex-A75](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

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 `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.
- Dual ACE masters.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. The following models also exist, with a fixed number of cores per cluster:

- ARMCortexA75x1CT.
- ARMCortexA75x2CT.
- ARMCortexA75x3CT.
- ARMCortexA75x4CT.

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 4-125: 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>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.

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.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[4]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[4]	PChannel	Slave	PChannels for cores
pmuirq[4]	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[4]	Signal	Slave	Individual processor RAM Error Interrupt signal input.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core virtual System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtual FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtual IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-A75

cpu0.AA64nAA32

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

Type: bool. Default value: 0x1.

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CFGTE

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

Type: `bool`. Default value: `0x0`.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-write_access_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-write_latency

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.

Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).

Type: int. Default value: 0x100.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.
Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: `bool`. Default value: `0x0`.

cpu0.VINITHI

Reset value of SCTLR.V.
Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A75

BROADCASTATOMIC

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.
Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.
Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CCSIDR-L1D_override

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

Type: `int`. Default value: `0x0`.

CCSIDR-L1I_override

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

Type: `int`. Default value: `0x0`.

CCSIDR-L2_override

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`dcache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-prefetch_enabled`

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

Type: `bool`. Default value: `0x0`.

`dcache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-state_modelled`

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

`dcache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: bool. Default value: 0x0.

icache-read_access_latency

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.

Type: int. Default value: 0x0.

icache-read_latency

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.

Type: int. Default value: 0x0.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: bool. Default value: 0x0.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

l3cache-miss_latency

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.

Type: int. Default value: 0x0.

l3cache-read_access_latency

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.

Type: int. Default value: 0x0.

l3cache-read_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.
Type: `int`. Default value: `0x80000`.

l3cache-snoop_data_transfer_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.
Type: `int`. Default value: `0x0`.

NUM_CORES

Number of cores per cluster.
Type: `int`. Default value: `0x1`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.
Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).
Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).
Type: `int`. Default value: `0x0`.

ptw_latency

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

tlb_latency

TLB latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

use_architectural_names

Use names SP/LR/PC instead of R13/R14/R15.
Type: `bool`. Default value: `0x0`.

walk_cache_latency

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

4.5.27 ARM Cortex A76 AECT

ARM Cortex A76 AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-126: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM Cortex A76 AECT contains the following CADI targets:

- `ARM_Cortex-A76AE`
- `Cluster_ARM_Cortex-A76AE`
- `PVCache`
- `TlbCadi`

ARM Cortex A76 AECT contains the following MTI components:

- [ARM_Cortex-A76AE](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

About ARMCortexA76AECT

The model supports the following features:

- DynamIQ™ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, or `nPMBIRQ` signals.
- DynamIQ™ features that are irrelevant to the programmers' view simulation, for example:
 - Automatic CPU retention mode.
 - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.
- Split/Lock

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. The following models also exist, with a fixed number of cores per cluster:

- ARMCortexA76AEx2CT.
- ARMCortexA76AEx4CT.
- ARMCortexA76AEx6CT.
- ARMCortexA76AEx8CT.

The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-7).

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](#).

AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following caveats:

- 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 4-127: 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.
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.

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 ARM_Cortex-A76AE

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.
Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.
Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-size`

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

`cpu0.l2cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-snoop_issue_latency`

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: int. Default value: 0xf000000.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: int. Default value: 0x3c.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A76AE

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

`cpi_div`

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

`cpi_mul`

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

`dcache-hit_latency`

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`dcache-maintenance_latency`

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`dcache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-prefetch_enabled`

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

Type: `bool`. Default value: `0x0`.

`dcache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_lock_step

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

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

has_statistical_profiling

Whether Statistical Based Profiling is implemented (FEAT_SPE).

Type: `bool`. Default value: `0x1`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

`icache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`icache-state_modelled`

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

`l3cache-hit_latency`

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`l3cache-maintenance_latency`

L3 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`l3cache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-size`

L3 Cache size in bytes.

Type: `int`. Default value: `0x100000`.

`l3cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-snoop_issue_latency`

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`l3cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`NUM_CORES`

Number of cores per cluster.

Type: `int`. Default value: `0x2`.

`pchannel_treat_simreset_as_poreset`

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

`periph_address_end`

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

`periph_address_start`

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

`ptw_latency`

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

Type: `int`. Default value: `0x0`.

`tlb_latency`

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

Type: `int`. Default value: `0x0`.

`treat-dcache-cmos-to-pou-as-nop`

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.

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.28 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 4-128: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA76CT contains the following CADI targets:

- ARM_Cortex-A76
- Cluster_ARM_Cortex-A76
- PVCache
- TlbCadi

ARMCortexA76CT contains the following MTI components:

- [ARM_Cortex-A76](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

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 component has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. The following components also exist, with a fixed number of cores per cluster:

- ARMCortexA76x1CT.
- ARMCortexA76x2CT.
- ARMCortexA76x3CT.
- ARMCortexA76x4CT.

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 ARMCortexA76CT

Table 4-129: 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.
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.

Name	Protocol	Type	Description
dbgnopwrdown[4]	Signal	Master	These signals relate to core power down.
dbgpwrupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[4]	PChannel	Slave	PChannels for cores
pmbirq[4]	Signal	Master	Interrupt signal from statistical profiling unit.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core virtual System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
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.

Name	Protocol	Type	Description
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-A76

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CFGTE

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-miss_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_access_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_latency

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. Type: int. Default value: 0x0.

cpu0.l2cache-size

L2 Cache size in bytes. Type: int. Default value: 0x80000.

cpu0.l2cache-snoop_data_transfer_latency

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. Type: int. Default value: 0x0.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true. Type: int. Default value: 0x0.

cpu0.l2cache-write_access_latency

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. Type: int. Default value: 0x0.

cpu0.l2cache-write_latency

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. Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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). Type: int. Default value: 0x100.

cpu0.min_sync_level

Force minimum syncLevel (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll). Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register. Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls. Type: int. Default value: 0x000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf0000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf0000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A76

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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.

Type: int. Default value: 0x0.

default_opmode

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.

Type: int. Default value: 0x4.

diagnostics

Enable DynamIQ diagnostic messages.

Type: bool. Default value: 0x0.

enable_simulation_performance_optimizations

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).

Type: bool. Default value: 0x1.

ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: bool. Default value: 0x0.

ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: bool. Default value: 0x0.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: bool. Default value: 0x0.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: bool. Default value: 0x0.

force_zero_PSTATE_PAN

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.

Type: bool. Default value: 0x0.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: bool. Default value: 0x1.

has_acp

If true, Accelerator Coherency Port is configured.

Type: bool. Default value: 0x0.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

has_statistical_profiling

Whether Statistical Based Profiling is implemented (FEAT_SPE).

Type: `bool`. Default value: `0x1`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x100000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

treat-dcache-cmos-to-pou-as-nop

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.

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.29 ARM CortexA77CT

ARM CortexA77CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-130: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM CortexA77CT contains the following CADI targets:

- ARM_Cortex-A77
- Cluster_ARM_Cortex-A77
- PVCache
- TlbCadi

ARMCortexA77CT contains the following MTI components:

- [ARM_Cortex-A77](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

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 `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 following models also exist, with a fixed number of cores per cluster:

- ARMCortexA77x1CT.
- ARMCortexA77x2CT.
- ARMCortexA77x3CT.
- ARMCortexA77x4CT.

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 ARMCortexA77CT

Table 4-131: 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.

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 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 ARM_Cortex-A77

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.
Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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.
Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).
Type: int. Default value: 0x100.

cpu0.min_sync_level

Force minimum syncLevel (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll).
Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.
Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.
Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.
Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.
Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

`cpu0.VINITI`

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A77

BROADCASTATOMIC

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAIN

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

`dcache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-state_modelled`

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

`dcache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`default_opmode`

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.

Type: `int`. Default value: `0x4`.

`diagnostics`

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

`enable_simulation_performance_optimizations`

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).

Type: `bool`. Default value: `0x1`.

`ext_abort_device_read_is_sync`

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`. Default value: `0x0`.

`ext_abort_device_write_is_sync`

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

has_statistical_profiling

Whether Statistical Based Profiling is implemented (FEAT_SPE).

Type: `bool`. Default value: `0x1`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x100000`.

`l3cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-snoop_issue_latency`

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`l3cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`NUM_CORES`

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

`pchannel_treat_simreset_as_poreset`

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

`periph_address_end`

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

`periph_address_start`

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

`ptw_latency`

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

Type: `int`. Default value: `0x0`.

`tlb_latency`

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

Type: `int`. Default value: `0x0`.

treat-dcache-cmos-to-pou-as-nop

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.

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.30 ARMCortexA78AECT

ARMCortexA78AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-132: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA78AECT contains the following CADI targets:

- `ARM_Cortex-A78AE`
- `Cluster_ARM_Cortex-A78AE`
- `PVCache`
- `TlbCadi`

ARMCortexA78AECT contains the following MTI components:

- [ARM_Cortex-A78AE](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)

- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

About ARMCortexA78AECT

The model supports the following features:

- DynamIQ™ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, or `nPMBIRQ` signals.
- DynamIQ™ features that are irrelevant to the programmers' view simulation, for example:
 - Automatic CPU retention mode.
 - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.
- 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

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. The following models also exist, with a fixed number of cores per cluster:

- ARMCortexA78AEx2CT.

- ARMCortexA78AEx4CT.

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](#).

AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following caveats:

- 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 4-133: 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

Name	Protocol	Type	Description
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[4]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset[4]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[4]	PChannel	Slave	PChannels for cores
pmbirq[4]	Signal	Master	Interrupt signal from statistical profiling unit.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.

Name	Protocol	Type	Description
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-A78AE

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CFGTE

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-maintenance_latency`

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-size`

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

`cpu0.l2cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-snoop_issue_latency`

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A78AE

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

`cpi_div`

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

`cpi_mul`

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

`dcache-hit_latency`

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`dcache-maintenance_latency`

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`dcache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-prefetch_enabled`

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

Type: `bool`. Default value: `0x0`.

`dcache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_lock_step

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

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_coherent_icache

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

has_statistical_profiling

Whether Statistical Based Profiling is implemented (FEAT_SPE).

Type: `bool`. Default value: `0x1`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: bool. Default value: 0x0.

icache-read_access_latency

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.

Type: int. Default value: 0x0.

icache-read_latency

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.

Type: int. Default value: 0x0.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: bool. Default value: 0x0.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

l3cache-miss_latency

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.

Type: int. Default value: 0x0.

l3cache-read_access_latency

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.

Type: int. Default value: 0x0.

l3cache-read_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.
Type: `int`. Default value: `0x100000`.

l3cache-snoop_data_transfer_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.
Type: `int`. Default value: `0x0`.

NUM_CORES

Number of cores per cluster.
Type: `int`. Default value: `0x2`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.
Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).
Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).
Type: `int`. Default value: `0x0`.

ptw_latency

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

treat-dcache-cmos-to-pou-as-nop

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.

Type: `int`. Default value: `0x0`.

treat_PAC_as_NOP

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.

Type: `bool`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.31 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 4-134: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA78CCT contains the following CADI targets:

- ARM_Cortex-A78C
- Cluster_ARM_Cortex-A78C
- PVCache
- TlbCadi

ARMCortexA78CCT contains the following MTI components:

- [ARM_Cortex-A78C](#)
- [ARMv8Cluster](#)

- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

About ARMCortexA78CCT

- 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 following models also exist, with a fixed number of cores per cluster:

- ARMCortexA78x1CCT.
- ARMCortexA78x2CCT.
- ARMCortexA78x4CCT.
- ARMCortexA78x6CCT.
- ARMCortexA78x8CCT.

The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-7).

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 ARMCortexA78CCT

Table 4-135: 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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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 ARM_Cortex-A78C

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.
Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.
Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks.

This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A78C

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

`cpi_div`

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

`cpi_mul`

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

`dcache-hit_latency`

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`dcache-maintenance_latency`

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`dcache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-prefetch_enabled`

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

Type: `bool`. Default value: `0x0`.

`dcache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`dcache-size`

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_coherent_icache

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`. Default value: `0x0`.

has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x100000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

`treat-dcache-cmos-to-pou-as-nop`

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.

Type: `int`. Default value: `0x0`.

`treat_PAC_as_NOP`

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.

Type: `bool`. Default value: `0x0`.

`use_architectural_names`

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

Type: `bool`. Default value: `0x0`.

`walk_cache_latency`

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

Type: `int`. Default value: `0x0`.

4.5.32 ARM Cortex A78CT

ARM Cortex A78CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-136: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM Cortex A78CT contains the following CADI targets:

- `ARM_Cortex-A78`
- `Cluster_ARM_Cortex-A78`
- `PVCache`
- `TlbCadi`

ARM Cortex A78CT contains the following MTI components:

- [ARM_Cortex-A78](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)

- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

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.

- 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 component has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. The following components also exist, with a fixed number of cores per cluster:

- ARMCortexA78x1CT.
- ARMCortexA78x2CT.
- ARMCortexA78x3CT.
- ARMCortexA78x4CT.

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 ARMCortexA78CT

Table 4-137: 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 <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.
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.

Name	Protocol	Type	Description
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug 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.
dbgprupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
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.

Name	Protocol	Type	Description
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-A78

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CFGTE

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks.

This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.VINITHI

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A78

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_coherent_icache

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`. Default value: `0x0`.

has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

`l3cache-size`

L3 Cache size in bytes.

Type: `int`. Default value: `0x100000`.

`l3cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-snoop_issue_latency`

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`l3cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`NUM_CORES`

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

`pchannel_treat_simreset_as_poreset`

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

`periph_address_end`

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

`periph_address_start`

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

`ptw_latency`

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

Type: `int`. Default value: `0x0`.

`tlb_latency`

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

Type: `int`. Default value: `0x0`.

treat-dcache-cmos-to-pou-as-nop

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.

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.33 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 4-138: IP revisions support

Revision	Quality level
r0p0	Full support
r1p3	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA510CT contains the following CADI targets:

- ARM_Cortex-A510
- Cluster_ARM_Cortex-A510
- PVCache
- TlbCadi

ARMCortexA510CT contains the following MTI components:

- [ARM_Cortex-A510](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PPUV1](#)
- [PVBusLogger](#)

- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

Parameters removed:

- `fault_unalign_to_unsupported_access`

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](#).

Limitations

This model does not support the Power Control State Machine (PCSM).

Ports for ARMCortexA510CT

Table 4-139: 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_powerdown_out	Signal	Master	An output from PPU that informs thermal controller of the core 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_powerdown_out[12]	Signal	Master	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.
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.

Name	Protocol	Type	Description
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 ARM_Cortex-A510

cpu0.CFGEND

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

Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_bus_width_in_bytes

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

Type: `int`. Default value: `0x10`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-ways

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

Type: int. Default value: 0x8.

cpu0.l2cache-write_access_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-write_bus_width_in_bytes

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

Type: int. Default value: 0x20.

cpu0.l2cache-write_latency

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.

Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).

Type: int. Default value: 0x100.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for Cluster_ARM_Cortex-A510

AEND0_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `int`. Default value: `0x0`.

AEND1_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `int`. Default value: `0x0`.

AEND2_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `int`. Default value: `0x0`.

AEND3_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `int`. Default value: `0x0`.

ASTART0_DEFAULT

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `int`. Default value: `0x0`.

ASTART1_DEFAULT

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `int`. Default value: `0x0`.

ASTART2_DEFAULT

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `int`. Default value: `0x0`.

ASTART3_DEFAULT

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `int`. Default value: `0x0`.

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x0`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x0`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

CPUCFR

Value of CPU Configuration Register.

Type: `int`. Default value: `0x0`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

`enable_simulation_performance_optimizations`

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).

Type: `bool`. Default value: `0x1`.

`ete.CLAIMTAGS`

Number of claim tags.

Type: `int`. Default value: `0x4`.

`ete.MAX_INST_PER_Q`

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

Type: `int`. Default value: `0x1`.

`ete.NumberOfRSPairs`

Number of resource selector pairs.

Type: `int`. Default value: `0x8`.

`ete.PIDR_CM0D`

TRCPIDR CM0D value.

Type: `int`. Default value: `0x0`.

`ete.PIDR_REVAND`

TRCPIDR REVAND value.

Type: `int`. Default value: `0x0`.

`ete.PIDR_REVISION`

TRCPIDR REVISION value.

Type: `int`. Default value: `0x0`.

`ete.Q_CADENCE`

Number of instruction blocks traced between two Q elements.

Type: `int`. Default value: `0x1`.

`ete.RES0_STATEFUL`

Whether RES0 bits are stateful or RAZ/WI.

Type: `bool`. Default value: `0x0`.

`ete.RETSTACK`

Return stack depth.

Type: `int`. Default value: `0x3`.

`ete.REVISION`

TRCIDR1 revision value.

Type: `int`. Default value: `0x0`.

`ete.SIM_OVERFLOW_GRANULARITY`

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

Type: `int`. Default value: `0x64`.

ete.SIM_OVERFLOW_PERCENTAGE

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

Type: `int`. Default value: `0x0`.

ete.SOURCE_ADDRESS

Allow generation of source address elements.

Type: `bool`. Default value: `0x0`.

ete.TRACE_OUTPUT

File to which to write trace byte stream.

Type: `string`. Default value: `""`.

ete.TRCRSRTA_FORCED_EXCEP

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`. Default value: `0x0`.

force_mte_tag_access_razwi_and_ignore_tag_checks

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.

Type: `bool`. Default value: `0x0`.

force_zero_mpam_partid_and_pmg

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.

Type: `int`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

has_ete

If true, implements the Embedded Trace Extension (FEAT_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-has_mpam

L3 Cache has MPAM support.

Type: `bool`. Default value: `0x1`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-ways

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

Type: `int`. Default value: `0x10`.

l3cache-write_access_latency

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.

Type: int. Default value: 0x0.

l3cache-write_latency

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.

Type: int. Default value: 0x0.

max_32bit_el

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: ELO supports A32. This parameter is ignored in Rev0 i.e. when revision_number = 0.

Type: int. Default value: 0x0.

memory_tagging_support_level

Equivalent to BROADCASTMTE.

Type: int. Default value: 0x2.

num_nodes

Number of transport nodes. Zero implies direct-connect configuration.

Type: int. Default value: 0x1.

patch_level

Cosmetic change to Revision field in MIDR/MIDR_EL1. Corresponds to the patch number Y in rXpY.

Type: int. Default value: 0x3.

ptw_latency

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

Type: int. Default value: 0x0.

ras_pfg_clock_mhz

RAS Pseudo-Fault generation clock rate in MHz.

Type: int. Default value: 0xc.

revision_number

Cosmetic change to Variant field in MIDR/MIDR_EL1. Corresponds to the revision number X in rXpY.

Type: int. Default value: 0x1.

tlb_latency

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

Type: int. Default value: 0x0.

treat_PAC_as_NOP

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.

Type: bool. Default value: 0x0.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.34 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 4-140: IP revisions support

Revision	Quality level
CortexA510 r0p0	Full support
CortexA510 r1p0	Full support
CortexA710 r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA510CT_CortexA710CT contains the following CADI targets:

- ARM_Cortex-A510
- ARM_Cortex-A710
- Cluster_ARM_Cortex510_CortexA710_Heterogeneous_Cluster
- PVCache

ARMCortexA510CT_CortexA710CT contains the following MTI components:

- [ARM_Cortex-A510](#)
- [ARM_Cortex-A710](#)
- [ARMv8Cluster](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PPUv1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `subcluster0.use_architectural_names`

Parameters removed:

- `subcluster0.fault_unalign_to_unsupported_access`

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:

- [4.5.33 ARMCortexA510CT](#) on page 622.
- [4.5.40 ARMCortexA710CT](#) on page 737.

Ports for ARMCortexA510CT_CortexA710CT

Table 4-141: Ports

Name	Protocol	Type	Description
<code>acp_s</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	-

Name	Protocol	Type	Description
AEND2MP	Value_64	Slave	-
AEND3MP	Value_64	Slave	-
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	-
ASTART2MP	Value_64	Slave	-
ASTART3MP	Value_64	Slave	-
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[22]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_powerdown_out	Signal	Master	An output from PPU that informs thermal controller of the core 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.
CNTHVSIQ[22]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[22]	Signal	Master	Timer signals to SOC.
CNTPSIQ[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_powerdown_out[22]	Signal	Master	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.

Name	Protocol	Type	Description
cti[22]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[22]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[22]	Signal	Master	No power-down request.
dbgpwrupreq[22]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[22]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[22]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[22]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmuirq[22]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	Cluster wake request signal.
ppu_core_irq[22]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[22]	Signal	Slave	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.

Name	Protocol	Type	Description
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	UtilityBus, through which external component can access in-cluster component and cluster control registers
vcpumntirq[22]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[22]	Signal	Slave	Virtualised FIQ.
virq[22]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L3 system and becomes coherent (assuming attributes are set correctly).
vsei[22]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-A510

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-miss_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_bus_width_in_bytes`

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

Type: `int`. Default value: `0x10`.

`cpu0.l2cache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-size`

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

`cpu0.l2cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-snoop_issue_latency`

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-ways`

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

Type: `int`. Default value: `0x8`.

`cpu0.l2cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_bus_width_in_bytes`

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

Type: `int`. Default value: `0x20`.

`cpu0.l2cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.RVBARADDR`

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for ARM_Cortex-A710**cpu0.CFGEND**

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

Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-size`

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

`cpu0.l2cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-snoop_issue_latency`

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.RVBARADDR`

Value of `RVBAR_ELx` register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex510_CortexA710_Heterogeneous_Cluster

AEND0_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `int`. Default value: `0x0`.

AEND1_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `int`. Default value: `0x0`.

AEND2_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `int`. Default value: `0x0`.

AEND3_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `int`. Default value: `0x0`.

ASTART0_DEFAULT

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `int`. Default value: `0x0`.

ASTART1_DEFAULT

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `int`. Default value: `0x0`.

ASTART2_DEFAULT

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `int`. Default value: `0x0`.

ASTART3_DEFAULT

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `int`. Default value: `0x0`.

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-has_mpam

L3 Cache has MPAM support.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x0`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

num_nodes

Number of transport nodes. Zero implies direct-connect configuration.

Type: `int`. Default value: `0x1`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

4.5.35 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 4-142: IP revisions support

Revision	Quality level
CortexA510 r0p0	Full support
CortexA710 r0p0	Full support
CortexX3 r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA510CT_CortexA710CT_CortexX3CT contains the following CADI targets:

- ARM_Cortex-A510
- ARM_Cortex-A710
- ARM_Cortex-X3
- Cluster_ARM_CortexA510_CortexA710_CortexX3_Heterogeneous_Cluster
- PVCache

ARMCortexA510CT_CortexA710CT_CortexX3CT contains the following MTI components:

- [ARM_Cortex-A510](#)
- [ARM_Cortex-A710](#)
- [ARM_Cortex-X3](#)
- [ARMv8Cluster](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PPUv1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `subcluster0.use_architectural_names`

Parameters removed:

- `subcluster0.fault_unalign_to_unsupported_access`
- `subcluster2.advsimd_bf16_support_level`
- `subcluster2.fault_unalign_to_unsupported_access`
- `subcluster2.has_16bit_vmids`

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:

- [4.5.33 ARMCortexA510CT](#) on page 622.
- [4.5.40 ARMCortexA710CT](#) on page 737.
- [4.5.64 ARMCortexX3CT](#) on page 990.

Ports for ARMCortexA510CT_CortexA710CT_CortexX3CT

Table 4-143: 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_powerdown_out	Signal	Master	An output from PPU that informs thermal controller of the core 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.
CNTHVSIQ[30]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[30]	Signal	Master	Timer signals to SOC.
CNTPSIQ[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_powerdown_out[30]	Signal	Master	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.
pmuirq[30]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[30]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[30]	Signal	Slave	PPU Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[30]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[30]	Value_64	Slave	Reset vector base address.

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[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_Cortex-A510

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-miss_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_bus_width_in_bytes

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

Type: `int`. Default value: `0x10`.

cpu0.l2cache-read_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-ways

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

Type: `int`. Default value: `0x8`.

cpu0.l2cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_bus_width_in_bytes

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

Type: `int`. Default value: `0x20`.

cpu0.l2cache-write_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for ARM_Cortex-A710**cpu0.CFGEND**

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

Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

Parameters for ARM_Cortex-X3**cpu0.CFGEND**

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

Type: `bool`. Default value: `0x0`.

cpu0.CFGTE

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

Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.force-fpsid

Override the FPSID value.

Type: `bool`. Default value: `0x1`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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. Type: int. Default value: 0x0.

cpu0.l2cache-size

L2 Cache size in bytes. Type: int. Default value: 0x100000.

cpu0.l2cache-snoop_data_transfer_latency

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. Type: int. Default value: 0x0.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true. Type: int. Default value: 0x0.

cpu0.l2cache-write_access_latency

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. Type: int. Default value: 0x0.

cpu0.l2cache-write_latency

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. Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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). Type: int. Default value: 0x100.

cpu0.min_sync_level

Force minimum syncLevel (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll). Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register. Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls. Type: int. Default value: 0x000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

`cpu0.vfp-present`

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for Cluster_ARM_CortexA510_CortexA710_CortexX3_Heterogeneous_Cluster

AEND0_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `int`. Default value: `0x0`.

AEND1_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `int`. Default value: `0x0`.

AEND2_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `int`. Default value: `0x0`.

AEND3_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `int`. Default value: `0x0`.

ASTART0_DEFAULT

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `int`. Default value: `0x0`.

ASTART1_DEFAULT

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `int`. Default value: `0x0`.

ASTART2_DEFAULT

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `int`. Default value: `0x0`.

ASTART3_DEFAULT

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `int`. Default value: `0x0`.

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-has_mpam

L3 Cache has MPAM support.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x0`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

num_nodes

Number of transport nodes. Zero implies direct-connect configuration.

Type: `int`. Default value: `0x1`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

4.5.36 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 4-144: IP revisions support

Revision	Quality level
CortexA510 r0p0	Full support
CortexA715 r0p0	Full support
CortexX3 r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA510CT_CortexA715CT_CortexX3CT contains the following CADI targets:

- ARM_Cortex-A510
- ARM_Cortex-A715
- ARM_Cortex-X3
- Cluster_ARM_CortexA510_CortexA715_CortexX3_Heterogeneous_Cluster
- PVCache

ARMCortexA510CT_CortexA715CT_CortexX3CT contains the following MTI components:

- [ARM_Cortex-A510](#)
- [ARM_Cortex-A715](#)
- [ARM_Cortex-X3](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PPUv1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Ports added:

- `pmbirq`

Parameters added:

- `subcluster0.use_architectural_names`

Parameters removed:

- `subcluster0.fault_unalign_to_unsupported_access`
- `subcluster1.advsimd_bf16_support_level`
- `subcluster1.fault_unalign_to_unsupported_access`
- `subcluster1.has_16bit_vmids`
- `subcluster2.advsimd_bf16_support_level`
- `subcluster2.fault_unalign_to_unsupported_access`
- `subcluster2.has_16bit_vmids`

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:

- [4.5.33 ARMCortexA510CT](#) on page 622.
- [4.5.41 ARMCortexA715CT](#) on page 752.
- [4.5.64 ARMCortexX3CT](#) on page 990.

Ports for ARMCortexA510CT_CortexA715CT_CortexX3CT

Table 4-145: 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_powerdown_out	Signal	Master	An output from PPU that informs thermal controller of the core 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

Name	Protocol	Type	Description
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_powerdown_out[30]	Signal	Master	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.

Name	Protocol	Type	Description
ppu_core_irq[30]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[30]	Signal	Slave	PPU Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[30]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[30]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[30]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[30]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[30]	Signal	Master	Interrupt signal from the trace buffer unit.
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_Cortex-A510

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_bus_width_in_bytes

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

Type: `int`. Default value: `0x10`.

cpu0.l2cache-read_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-ways

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

Type: `int`. Default value: `0x8`.

cpu0.l2cache-write_access_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-write_bus_width_in_bytes

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

Type: int. Default value: 0x20.

cpu0.l2cache-write_latency

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.

Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).

Type: int. Default value: 0x100.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: int. Default value: 0xf000000.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: int. Default value: 0x3c.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: int. Default value: 0xab.

cpu0.trace_special_hlt_imm16

For this HLT number, IF enable_trace_special_hlt_imm16=true, skip performing usual HLT execution but call MTI trace if registered.
Type: int. Default value: 0xf000.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: bool. Default value: 0x0.

cpu0.vfp-present

Set whether the model has VFP support.
Type: bool. Default value: 0x1.

Parameters for ARM_Cortex-A715

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.force-fpsid

Override the FPSID value.

Type: bool. Default value: 0x1.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-miss_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_access_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: int. Default value: 0x80000.

cpu0.l2cache-snoop_data_transfer_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-write_access_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-write_latency

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.

Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).

Type: int. Default value: 0x100.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.
Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.
Type: `bool`. Default value: `0x1`.

Parameters for ARM_Cortex-X3**cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: `bool`. Default value: `0x0`.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.
Type: `bool`. Default value: `0x0`.

cpu0.force-fpsid

Override the FPSID value.

Type: `bool`. Default value: `0x1`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x100000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.RVBARADDR`

Value of `RVBAR_ELx` register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf0000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for**Cluster_ARM_CortexA510_CortexA715_CortexX3_Heterogeneous_Cluster****AEND0_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `int`. Default value: `0x0`.

AEND1_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `int`. Default value: `0x0`.

AEND2_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `int`. Default value: `0x0`.

AEND3_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `int`. Default value: `0x0`.

ASTART0_DEFAULT

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `int`. Default value: `0x0`.

ASTART1_DEFAULT

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `int`. Default value: `0x0`.

ASTART2_DEFAULT

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `int`. Default value: `0x0`.

ASTART3_DEFAULT

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `int`. Default value: `0x0`.

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x0`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x0`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-has_mpam

L3 Cache has MPAM support.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-ways

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

Type: `int`. Default value: `0x10`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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.
Type: int. Default value: 0x0.

num_nodes

Number of transport nodes. Zero implies direct-connect configuration.
Type: int. Default value: 0x1.

4.5.37 ARMCortexA520CT

ARMCortexA520CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-146: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA520CT contains the following CADI targets:

- ARM_Cortex-A520
- Cluster_ARM_Cortex-A520
- PVCache
- TlbCadi

ARMCortexA520CT contains the following MTI components:

- [ARM_Cortex-A520](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PPUv1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters removed:

- `fault_unalign_to_unsupported_access`
- `has_trbe`

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 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

This model does not support the Power Control State Machine (PCSM).

Ports for ARMCortexA520CT

Table 4-147: 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_powerdown_out	Signal	Master	An output from PPUs that informs thermal controller of the core 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_powerdown_out[14]	Signal	Master	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.

Name	Protocol	Type	Description
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 ARM_Cortex-A520

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-miss_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_access_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_bus_width_in_bytes

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

Type: int. Default value: 0x10.

cpu0.l2cache-read_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: int. Default value: 0x80000.

cpu0.l2cache-snoop_data_transfer_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-ways

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

Type: `int`. Default value: `0x8`.

`cpu0.l2cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_bus_width_in_bytes`

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

Type: `int`. Default value: `0x20`.

`cpu0.l2cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.RVBARADDR`

Value of `RVBAR_ELx` register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for Cluster_ARM_Cortex-A520

AEND0_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `int`. Default value: `0x0`.

AEND1_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `int`. Default value: `0x0`.

AEND2_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `int`. Default value: `0x0`.

AEND3_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `int`. Default value: `0x0`.

ASTART0_DEFAULT

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `int`. Default value: `0x0`.

ASTART1_DEFAULT

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `int`. Default value: `0x0`.

ASTART2_DEFAULT

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `int`. Default value: `0x0`.

ASTART3_DEFAULT

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `int`. Default value: `0x0`.

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x0`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x0`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_cache_protection

`core_cache_protection` can change `ERROFR`, `ERROPFGF` and `ERROPFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

CPUCFR

Value of CPU Configuration Register.

Type: `int`. Default value: `0x0`.

DBGROMADDR

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

Type: `int`. Default value: `0x0`.

DBGROMADDRV

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

Type: `bool`. Default value: `0x0`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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.

Type: int. Default value: 0x0.

enable_simulation_performance_optimizations

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).

Type: bool. Default value: 0x1.

ete.CLAIMTAGS

Number of claim tags.

Type: int. Default value: 0x4.

ete.MAX_INST_PER_Q

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

Type: int. Default value: 0x1.

ete.PIDR_CM0D

TRCPIDR CM0D value.

Type: int. Default value: 0x0.

ete.PIDR_REVAND

TRCPIDR REVAND value.

Type: int. Default value: 0x0.

ete.PIDR_REVISION

TRCPIDR REVISION value.

Type: int. Default value: 0x0.

ete.Q_CADENCE

Number of instruction blocks traced between two Q elements.

Type: int. Default value: 0x1.

ete.RES0_STATEFUL

Whether RES0 bits are stateful or RAZ/WI.

Type: bool. Default value: 0x0.

ete.RETSTACK

Return stack depth.

Type: int. Default value: 0x3.

ete.REVISION

TRCIDR1 revision value.

Type: int. Default value: 0x0.

ete.SIM_OVERFLOW_GRANULARITY

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

Type: int. Default value: 0x64.

ete.SIM_OVERFLOW_PERCENTAGE

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

Type: `int`. Default value: `0x0`.

ete.SOURCE_ADDRESS

Allow generation of source address elements.

Type: `bool`. Default value: `0x0`.

ete.TRACE_OUTPUT

File to which to write trace byte stream.

Type: `string`. Default value: `""`.

ete.TRCRSRTA_FORCED_EXCEP

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`. Default value: `0x0`.

force_mte_tag_access_razwi_and_ignore_tag_checks

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.

Type: `bool`. Default value: `0x0`.

force_zero_mpam_partid_and_pmg

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.

Type: `int`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_actlr2

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR_EL1[63:32].

Type: `bool`. Default value: `0x1`.

has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

has_ete

If true, implements the Embedded Trace Extension (FEAT_ETE). This option is discarded if ete plugin is explicitly loaded (`--plugin` or `-P`).

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-has_mpam

L3 Cache has MPAM support.

Type: `bool`. Default value: `0x1`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_bus_width_in_bytes

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

Type: `int`. Default value: `0x10`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-ways

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

Type: `int`. Default value: `0x10`.

`l3cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-write_bus_width_in_bytes`

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

Type: `int`. Default value: `0x10`.

`l3cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`memory_tagging_support_level`

Equivalent to `BROADCASTMTE`.

Type: `int`. Default value: `0x3`.

`num_acp`

Number of ACP ports.

Type: `int`. Default value: `0x0`.

`NUM_CORES`

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

`num_nodes`

Number of transport nodes. Zero implies direct-connect configuration.

Type: `int`. Default value: `0x1`.

`pmu-num_counters`

Number of PMU counters implemented.

Type: `int`. Default value: `0x6`.

`ptw_latency`

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

Type: `int`. Default value: `0x0`.

`ras_extra_configurations`

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}].

Type: string. Default value: "[{ \"Index\": 1, \"ERXMISCO_mask\": 0xFFFFc0003fc3, \"ERXMISC1_mask\": 0x03F870003FF30f07, \"ERXPGCTL_reset\": 0x1000 }, { \"Index\": 2, \"ERXMISCO_mask\": 0xFFFFe007ffc0, \"ERXMISCO_reset\": 0x2, \"ERXSTATUS_IERR_mask\": 0x300 , \"ERXMISC1_mask\": 0x0FF8700fFFF31f0f, \"ERXPGCTL_reset\": 0x1000 }]\".

ras_pfg_clock_mhz

RAS Pseudo-Fault generation clock rate in MHz.

Type: int. Default value: 0xc.

revision_number

Cosmetic change to Variant field in MIDR/MIDR_EL1. Corresponds to the revision number X in rXpY.

Type: int. Default value: 0x0.

tlb_latency

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

Type: int. Default value: 0x0.

treat_PAC_as_NOP

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.

Type: bool. Default value: 0x0.

walk_cache_latency

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

Type: int. Default value: 0x0.

4.5.38 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 4-148: IP revisions support

Revision	Quality level
CortexA520 r0p0	Full support
CortexA720 r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA520CT_CortexA720CT contains the following CADI targets:

- ARM_Cortex-A520
- ARM_Cortex-A720
- Cluster_ARM_CortexA520_CortexA720_Heterogeneous_Cluster

- PVCache

ARMCortexA520CT_CortexA720CT contains the following MTI components:

- [ARM_Cortex-A520](#)
- [ARM_Cortex-A720](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PPUv1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters removed:

- `has_trbe`
- `subcluster0.fault_unalign_to_unsupported_access`
- `subcluster1.advsimd_bf16_support_level`
- `subcluster1.fault_unalign_to_unsupported_access`
- `subcluster1.has_16bit_vmids`

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:

- [4.5.37 ARMCortexA520CT](#) on page 686.
- [4.5.42 ARMCortexA720CT](#) on page 769.

Ports for ARMCortexA520CT_CortexA720CT

Table 4-149: Ports

Name	Protocol	Type	Description
<code>acp_s[2]</code>	PVBus	Slave	AXI ACP slave port
<code>AEND0MP</code>	Value_64	Slave	DynamiQ port to obtain end address of valid peripheral address range (exclusive).
<code>AEND1MP</code>	Value_64	Slave	-
<code>AEND2MP</code>	Value_64	Slave	-
<code>AEND3MP</code>	Value_64	Slave	-
<code>ASTART0MP</code>	Value_64	Slave	ASTART and AEND ports are captured during reset only. also if a range is invalid (ASTART >= AEND), then params are used DynamiQ port to obtain start address of valid peripheral address range (inclusive).
<code>ASTART1MP</code>	Value_64	Slave	-
<code>ASTART2MP</code>	Value_64	Slave	-
<code>ASTART3MP</code>	Value_64	Slave	-
<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[26]</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>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>cluster_powerdown_out</code>	Signal	Master	An output from PPU's that informs thermal controller of the core power info

Name	Protocol	Type	Description
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:22] 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_powerdown_out[26]	Signal	Master	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.

Name	Protocol	Type	Description
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[26]	Signal	Master	Interrupt signal from 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	Cluster wake request signal.
ppu_core_irq[26]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[26]	Signal	Slave	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[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	UtilityBus, through which external component can access in-cluster component and cluster control registers
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 L3 system and becomes coherent (assuming attributes are set correctly).
vsei[26]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-A520

cpu0.CFGEND

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

Type: `bool`. Default value: `0x0`.

`cpu0.CRYPTODISABLE`

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

`cpu0.enable_trace_special_hlt_imm16`

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

`cpu0.l2cache-hit_latency`

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-maintenance_latency`

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_bus_width_in_bytes`

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

Type: `int`. Default value: `0x10`.

`cpu0.l2cache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-size`

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

`cpu0.l2cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-ways

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

Type: int. Default value: 0x8.

cpu0.l2cache-write_access_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-write_bus_width_in_bytes

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

Type: int. Default value: 0x20.

cpu0.l2cache-write_latency

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.

Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).

Type: int. Default value: 0x100.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for ARM_Cortex-A720

cpu0.CFGEND

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

Type: `bool`. Default value: `0x0`.

cpu0.CFGTE

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.force-fpsid

Override the FPSID value.

Type: bool. Default value: 0x1.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-miss_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_access_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: int. Default value: 0x80000.

cpu0.l2cache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for Cluster_ARM_CortexA520_CortexA720_Heterogeneous_Cluster

AEND0_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `int`. Default value: `0x0`.

AEND1_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `int`. Default value: `0x0`.

AEND2_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `int`. Default value: `0x0`.

AEND3_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `int`. Default value: `0x0`.

ASTART0_DEFAULT

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `int`. Default value: `0x0`.

ASTART1_DEFAULT

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `int`. Default value: `0x0`.

ASTART2_DEFAULT

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `int`. Default value: `0x0`.

ASTART3_DEFAULT

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `int`. Default value: `0x0`.

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x0`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x0`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

DBGROMADDR

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

Type: `int`. Default value: `0x0`.

DBGROMADDRV

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

Type: `bool`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-has_mpam

L3 Cache has MPAM support.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_bus_width_in_bytes

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

Type: `int`. Default value: `0x10`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l3cache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

l3cache-ways

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

Type: `int`. Default value: `0x10`.

l3cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

l3cache-write_bus_width_in_bytes

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

Type: `int`. Default value: `0x10`.

l3cache-write_latency

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.

Type: `int`. Default value: `0x0`.

num_nodes

Number of transport nodes. Zero implies direct-connect configuration.

Type: `int`. Default value: `0x1`.

revision_number

Cosmetic change to Variant field in MIDR/MIDR_EL1. Corresponds to the revision number X in rXpY.

Type: `int`. Default value: `0x0`.

4.5.39 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 4-150: IP revisions support

Revision	Quality level
CortexA520 r0p0	Full support

Revision	Quality level
CortexA720 r0p0	Preliminary support
CortexX4 r0p0	Preliminary support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA520CT_CortexA720CT_CortexX4CT contains the following CADI targets:

- ARM_Cortex-A520
- ARM_Cortex-A720
- ARM_Cortex-X4
- Cluster_ARM_CortexA520_CortexA720_CortexX4_Heterogeneous_Cluster
- PVCache

ARMCortexA520CT_CortexA720CT_CortexX4CT contains the following MTI components:

- [ARM_Cortex-A520](#)
- [ARM_Cortex-A720](#)
- [ARM_Cortex-X4](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PPUv1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `subcluster2.cpu0.l2cache-hit_latency`
- `subcluster2.cpu0.l2cache-maintenance_latency`
- `subcluster2.cpu0.l2cache-miss_latency`
- `subcluster2.cpu0.l2cache-read_access_latency`
- `subcluster2.cpu0.l2cache-read_latency`
- `subcluster2.cpu0.l2cache-snoop_data_transfer_latency`
- `subcluster2.cpu0.l2cache-snoop_issue_latency`

- subcluster2.cpu0.l2cache-write_access_latency
- subcluster2.cpu0.l2cache-write_latency
- subcluster2.cpu1.l2cache-hit_latency
- subcluster2.cpu1.l2cache-maintenance_latency
- subcluster2.cpu1.l2cache-miss_latency
- subcluster2.cpu1.l2cache-read_access_latency
- subcluster2.cpu1.l2cache-read_latency
- subcluster2.cpu1.l2cache-snoop_data_transfer_latency
- subcluster2.cpu1.l2cache-snoop_issue_latency
- subcluster2.cpu1.l2cache-write_access_latency
- subcluster2.cpu1.l2cache-write_latency
- subcluster2.cpu10.l2cache-hit_latency
- subcluster2.cpu10.l2cache-maintenance_latency
- subcluster2.cpu10.l2cache-miss_latency
- subcluster2.cpu10.l2cache-read_access_latency
- subcluster2.cpu10.l2cache-read_latency
- subcluster2.cpu10.l2cache-snoop_data_transfer_latency
- subcluster2.cpu10.l2cache-snoop_issue_latency
- subcluster2.cpu10.l2cache-write_access_latency
- subcluster2.cpu10.l2cache-write_latency
- subcluster2.cpu11.l2cache-hit_latency
- subcluster2.cpu11.l2cache-maintenance_latency
- subcluster2.cpu11.l2cache-miss_latency
- subcluster2.cpu11.l2cache-read_access_latency
- subcluster2.cpu11.l2cache-read_latency
- subcluster2.cpu11.l2cache-snoop_data_transfer_latency
- subcluster2.cpu11.l2cache-snoop_issue_latency
- subcluster2.cpu11.l2cache-write_access_latency
- subcluster2.cpu11.l2cache-write_latency
- subcluster2.cpu2.l2cache-hit_latency
- subcluster2.cpu2.l2cache-maintenance_latency
- subcluster2.cpu2.l2cache-miss_latency
- subcluster2.cpu2.l2cache-read_access_latency
- subcluster2.cpu2.l2cache-read_latency

- subcluster2.cpu2.l2cache-snoop_data_transfer_latency
- subcluster2.cpu2.l2cache-snoop_issue_latency
- subcluster2.cpu2.l2cache-write_access_latency
- subcluster2.cpu2.l2cache-write_latency
- subcluster2.cpu3.l2cache-hit_latency
- subcluster2.cpu3.l2cache-maintenance_latency
- subcluster2.cpu3.l2cache-miss_latency
- subcluster2.cpu3.l2cache-read_access_latency
- subcluster2.cpu3.l2cache-read_latency
- subcluster2.cpu3.l2cache-snoop_data_transfer_latency
- subcluster2.cpu3.l2cache-snoop_issue_latency
- subcluster2.cpu3.l2cache-write_access_latency
- subcluster2.cpu3.l2cache-write_latency
- subcluster2.cpu4.l2cache-hit_latency
- subcluster2.cpu4.l2cache-maintenance_latency
- subcluster2.cpu4.l2cache-miss_latency
- subcluster2.cpu4.l2cache-read_access_latency
- subcluster2.cpu4.l2cache-read_latency
- subcluster2.cpu4.l2cache-snoop_data_transfer_latency
- subcluster2.cpu4.l2cache-snoop_issue_latency
- subcluster2.cpu4.l2cache-write_access_latency
- subcluster2.cpu4.l2cache-write_latency
- subcluster2.cpu5.l2cache-hit_latency
- subcluster2.cpu5.l2cache-maintenance_latency
- subcluster2.cpu5.l2cache-miss_latency
- subcluster2.cpu5.l2cache-read_access_latency
- subcluster2.cpu5.l2cache-read_latency
- subcluster2.cpu5.l2cache-snoop_data_transfer_latency
- subcluster2.cpu5.l2cache-snoop_issue_latency
- subcluster2.cpu5.l2cache-write_access_latency
- subcluster2.cpu5.l2cache-write_latency
- subcluster2.cpu6.l2cache-hit_latency
- subcluster2.cpu6.l2cache-maintenance_latency
- subcluster2.cpu6.l2cache-miss_latency

- subcluster2.cpu6.l2cache-read_access_latency
- subcluster2.cpu6.l2cache-read_latency
- subcluster2.cpu6.l2cache-snoop_data_transfer_latency
- subcluster2.cpu6.l2cache-snoop_issue_latency
- subcluster2.cpu6.l2cache-write_access_latency
- subcluster2.cpu6.l2cache-write_latency
- subcluster2.cpu7.l2cache-hit_latency
- subcluster2.cpu7.l2cache-maintenance_latency
- subcluster2.cpu7.l2cache-miss_latency
- subcluster2.cpu7.l2cache-read_access_latency
- subcluster2.cpu7.l2cache-read_latency
- subcluster2.cpu7.l2cache-snoop_data_transfer_latency
- subcluster2.cpu7.l2cache-snoop_issue_latency
- subcluster2.cpu7.l2cache-write_access_latency
- subcluster2.cpu7.l2cache-write_latency
- subcluster2.cpu8.l2cache-hit_latency
- subcluster2.cpu8.l2cache-maintenance_latency
- subcluster2.cpu8.l2cache-miss_latency
- subcluster2.cpu8.l2cache-read_access_latency
- subcluster2.cpu8.l2cache-read_latency
- subcluster2.cpu8.l2cache-snoop_data_transfer_latency
- subcluster2.cpu8.l2cache-snoop_issue_latency
- subcluster2.cpu8.l2cache-write_access_latency
- subcluster2.cpu8.l2cache-write_latency
- subcluster2.cpu9.l2cache-hit_latency
- subcluster2.cpu9.l2cache-maintenance_latency
- subcluster2.cpu9.l2cache-miss_latency
- subcluster2.cpu9.l2cache-read_access_latency
- subcluster2.cpu9.l2cache-read_latency
- subcluster2.cpu9.l2cache-snoop_data_transfer_latency
- subcluster2.cpu9.l2cache-snoop_issue_latency
- subcluster2.cpu9.l2cache-write_access_latency
- subcluster2.cpu9.l2cache-write_latency

Parameters removed:

- `has_trbe`
- `subcluster0.fault_unalign_to_unsupported_access`
- `subcluster1.advsimd_bf16_support_level`
- `subcluster1.fault_unalign_to_unsupported_access`
- `subcluster1.has_16bit_vmids`
- `subcluster2.advsimd_bf16_support_level`

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:

- [4.5.37 ARMCortexA520CT](#) on page 686.
- [4.5.42 ARMCortexA720CT](#) on page 769.
- [4.5.65 ARMCortexX4CT](#) on page 1007.

Ports for ARMCortexA520CT_CortexA720CT_CortexX4CT

Table 4-151: Ports

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port
AEND0MP	Value_64	Slave	DynamlQ port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	-
AEND2MP	Value_64	Slave	-
AEND3MP	Value_64	Slave	-
ASTART0MP	Value_64	Slave	ASTART and AEND ports are captured during reset only. also if a range is invalid (ASTART >= AEND), then params are used DynamlQ port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	-
ASTART2MP	Value_64	Slave	-
ASTART3MP	Value_64	Slave	-
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_powerdown_out	Signal	Master	An output from PPU that informs thermal controller of the core 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.
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.

Name	Protocol	Type	Description
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_powerdown_out[36]	Signal	Master	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.
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	Cluster wake request signal.
ppu_core_irq[36]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[36]	Signal	Slave	Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.

Name	Protocol	Type	Description
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[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	UtilityBus, through which external component can access in-cluster component and cluster control registers
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 L3 system and becomes coherent (assuming attributes are set correctly).
vsei[36]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-A520

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.
Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.
Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_bus_width_in_bytes

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

Type: `int`. Default value: `0x10`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-ways

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

Type: `int`. Default value: `0x8`.

cpu0.l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_bus_width_in_bytes

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

Type: `int`. Default value: `0x20`.

cpu0.l2cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Value of `RVBAR_ELx` register.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

`cpu0.vfp-present`

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for ARM_Cortex-A720

`cpu0.CFGEND`

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

Type: `bool`. Default value: `0x0`.

`cpu0.CFGTE`

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

Type: `bool`. Default value: `0x0`.

`cpu0.CRYPTODISABLE`

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

`cpu0.enable_trace_special_hlt_imm16`

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

`cpu0.force-fpsid`

Override the FPSID value.

Type: `bool`. Default value: `0x1`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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.
Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).
Type: int. Default value: 0x100.

cpu0.min_sync_level

Force minimum syncLevel (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll).
Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.
Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.
Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.
Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.
Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

`cpu0.vfp-present`

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for ARM_Cortex-X4

`cpu0.CFGEND`

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

Type: `bool`. Default value: `0x0`.

`cpu0.CFGTE`

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

Type: `bool`. Default value: `0x0`.

`cpu0.CRYPTODISABLE`

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

`cpu0.enable_trace_special_hlt_imm16`

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

`cpu0.l2cache-hit_latency`

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.
Type: int. Default value: 0x0.

cpu0.l2cache-miss_latency

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.
Type: int. Default value: 0x0.

cpu0.l2cache-read_access_latency

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.
Type: int. Default value: 0x0.

cpu0.l2cache-read_latency

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.
Type: int. Default value: 0x0.

cpu0.l2cache-size

L2 Cache size in bytes.
Type: int. Default value: 0x80000.

cpu0.l2cache-snoop_data_transfer_latency

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.
Type: int. Default value: 0x0.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.
Type: int. Default value: 0x0.

cpu0.l2cache-ways

L2 Cache number of ways (sets are implicit from size).
Type: int. Default value: 0x8.

cpu0.l2cache-write_access_latency

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.
Type: int. Default value: 0x0.

cpu0.l2cache-write_latency

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.
Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).
Type: int. Default value: 0x100.

cpu0.min_sync_level

Force minimum syncLevel (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll).
Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.
Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.
Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.
Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.
Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

`cpu0.vfp-present`

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for Cluster_ARM_CortexA520_CortexA720_CortexX4_Heterogeneous_Cluster

`AEND0_DEFAULT`

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `int`. Default value: `0x0`.

`AEND1_DEFAULT`

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `int`. Default value: `0x0`.

`AEND2_DEFAULT`

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `int`. Default value: `0x0`.

`AEND3_DEFAULT`

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `int`. Default value: `0x0`.

ASTART0_DEFAULT

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `int`. Default value: `0x0`.

ASTART1_DEFAULT

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `int`. Default value: `0x0`.

ASTART2_DEFAULT

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `int`. Default value: `0x0`.

ASTART3_DEFAULT

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `int`. Default value: `0x0`.

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x0`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x0`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

DBGROMADDR

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

Type: `int`. Default value: `0x0`.

DBGROMADDRV

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

Type: `bool`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-has_mpam

L3 Cache has MPAM support.

Type: `bool`. Default value: `0x1`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_bus_width_in_bytes

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

Type: `int`. Default value: `0x10`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-ways

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

Type: `int`. Default value: `0x10`.

l3cache-write_access_latency

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.

Type: `int`. Default value: 0x0.

l3cache-write_bus_width_in_bytes

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

Type: `int`. Default value: 0x10.

l3cache-write_latency

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.

Type: `int`. Default value: 0x0.

num_nodes

Number of transport nodes. Zero implies direct-connect configuration.

Type: `int`. Default value: 0x1.

revision_number

Cosmetic change to Variant field in MIDR/MIDR_EL1. Corresponds to the revision number X in rXpY.

Type: `int`. Default value: 0x0.

4.5.40 ARM Cortex A710CT

ARM Cortex A710CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-152: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM Cortex A710CT contains the following CADI targets:

- ARM_Cortex-A710
- Cluster_ARM_Cortex-A710
- PVCache
- TlbCadi

ARM Cortex A710CT contains the following MTI components:

- [ARM_Cortex-A710](#)
- [ARMv8Cluster](#)
- [DSU](#)

- [GICv3CPUInterfaceDecoder](#)
- [PPUV1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

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 ELO.

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](#).

Limitations

This model does not support the Power Control State Machine (PCSM).

Ports for ARMCortexA710CT

Table 4-153: 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_powerdown_out	Signal	Master	An output from PPUs that informs thermal controller of the core 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

Name	Protocol	Type	Description
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_powerdown_out[12]	Signal	Master	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.
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.

Name	Protocol	Type	Description
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 ARM_Cortex-A710

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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.
Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).
Type: int. Default value: 0x100.

cpu0.min_sync_level

Force minimum syncLevel (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll).
Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.
Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.
Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.
Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.
Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-A710

`AEND0_DEFAULT`

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `int`. Default value: `0x0`.

`AEND1_DEFAULT`

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `int`. Default value: `0x0`.

`AEND2_DEFAULT`

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `int`. Default value: `0x0`.

`AEND3_DEFAULT`

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `int`. Default value: `0x0`.

`ASTART0_DEFAULT`

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `int`. Default value: `0x0`.

ASTART1_DEFAULT

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `int`. Default value: `0x0`.

ASTART2_DEFAULT

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `int`. Default value: `0x0`.

ASTART3_DEFAULT

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `int`. Default value: `0x0`.

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

ete.CLAIMTAGS

Number of claim tags.

Type: `int`. Default value: `0x20`.

ete.MAX_INST_PER_Q

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

Type: `int`. Default value: `0x1`.

ete.NumberOfRSPairs

Number of resource selector pairs.

Type: `int`. Default value: `0x8`.

ete.PIDR_CM0D

TRCPIDR CM0D value.

Type: `int`. Default value: `0x0`.**ete.PIDR_REVAND**

TRCPIDR REVAND value.

Type: `int`. Default value: `0x1`.**ete.PIDR_REVISION**

TRCPIDR REVISION value.

Type: `int`. Default value: `0x0`.**ete.Q_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `int`. Default value: `0x1`.**ete.RES0_STATEFUL**

Whether RES0 bits are stateful or RAZ/WI.

Type: `bool`. Default value: `0x0`.**ete.RETSTACK**

Return stack depth.

Type: `int`. Default value: `0x3`.**ete.REVISION**

TRCIDR1 revision value.

Type: `int`. Default value: `0x2`.**ete.SIM_OVERFLOW_GRANULARITY**

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

Type: `int`. Default value: `0x64`.**ete.SIM_OVERFLOW_PERCENTAGE**

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

Type: `int`. Default value: `0x0`.**ete.SOURCE_ADDRESS**

Allow generation of source address elements.

Type: `bool`. Default value: `0x0`.**ete.TRACE_OUTPUT**

File to which to write trace byte stream.

Type: `string`. Default value: `""`.**ete.TRCSRSTA_FORCED_EXCEP**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`. Default value: `0x0`.**ext_abort_device_read_is_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`. Default value: `0x0`.

force_mte_tag_access_razwi_and_ignore_tag_checks

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.

Type: `bool`. Default value: `0x0`.

force_zero_mpam_partid_and_pmg

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.

Type: `int`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_ete

If true, implements the Embedded Trace Extension (FEAT_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x0`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

memory_tagging_support_level

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).

Type: `int`. Default value: `0x2`.

NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

num_nodes

Number of transport nodes. Zero implies direct-connect configuration.

Type: `int`. Default value: `0x1`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

treat-dcache-cmos-to-pou-as-nop

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.

Type: `int`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.41 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 4-154: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexA715CT contains the following CADI targets:

- ARM_Cortex-A715
- Cluster_ARM_Cortex-A715
- PVCache
- TlbCadi

ARMCortexA715CT contains the following MTI components:

- [ARM_Cortex-A715](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PPUv1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters removed:

- `advsimd_bf16_support_level`
- `fault_unalign_to_unsupported_access`
- `has_16bit_vmids`

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 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.
- This model does not support the Power Control State Machine (PCSM).

Ports for ARMCortexA715CT

Table 4-155: 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_powerdown_out	Signal	Master	An output from PPU that informs thermal controller of the core 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.

Name	Protocol	Type	Description
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_powerdown_out[12]	Signal	Master	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.
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.

Name	Protocol	Type	Description
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 ARM_Cortex-A715

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

`cpu0.enable_trace_special_hlt_imm16`

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

`cpu0.force-fpsid`

Override the FPSID value.

Type: `bool`. Default value: `0x1`.

`cpu0.l2cache-hit_latency`

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-maintenance_latency`

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-size`

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

`cpu0.l2cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-snoop_issue_latency`

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.RVBARADDR`

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

`cpu0.vfp-present`

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for Cluster_ARM_Cortex-A715

`AEND0_DEFAULT`

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `int`. Default value: `0x0`.

`AEND1_DEFAULT`

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `int`. Default value: `0x0`.

`AEND2_DEFAULT`

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `int`. Default value: `0x0`.

AEND3_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `int`. Default value: `0x0`.

ASTART0_DEFAULT

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `int`. Default value: `0x0`.

ASTART1_DEFAULT

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `int`. Default value: `0x0`.

ASTART2_DEFAULT

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `int`. Default value: `0x0`.

ASTART3_DEFAULT

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `int`. Default value: `0x0`.

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x0`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x0`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

CPUCFR

Value of CPU Configuration Register.

Type: `int`. Default value: `0x0`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.
Type: `int`. Default value: `0x8000`.

dcache-snoop_data_transfer_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.
Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.
Type: `int`. Default value: `0x0`.

default_opmode

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.
Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.
Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: bool. Default value: 0x1.

error_record_feature_register

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}

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".

Type: string. Default value:

```
"[{\"ED\":0x2,\"IMPDEF_3_2\":0x1,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,\"RP\":0x1,\"DUI\":0
```

```
"Visibility": "Cluster"},
```

```
{ "ED":0x2,"UI":0x2,"FI":0x2,"UE":0x1,"CFI":0x2,"CEC":0x2,"RP":0x1,"DUI":0x0,"CEO":0x0,"INJ":0
```

ete.CLAIMTAGS

Number of claim tags.

Type: `int`. Default value: `0x4`.

ete.MAX INST PER Q

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

Type: `int`. Default value: `0x1`.

```
ete.NumberOfRSPairs
```

Number of resource selector pairs.

Type: `int`. Default value: `0x8`.

```
ete.PIDR_CMOD
```

TRCPIDR CMOD value.

Type: `int`. Default value: `0x0`.

ete.PIDR REVAND

TRCPIDR REVAND value.

Type: `int`. Default value: `0x0`.

ete.PIDR REVISION

TRCPIDR REVISION value.

Type: `int`. Default value: `0x0`.

ete.Q CADENCE

Number of instruction blocks traced between two Q elements.

Type: `int`. Default value: `0x1`.

```
ete.RES0 STATEFUL
```

Whether RES0 bits are stateful or RAZ/WI.

Type: `bool`. Default value: `0x0`.

ete.RETSTACK

Return stack depth.

Type: `int`. Default value: `0x3`.

ete.REVISION

TRCIDR1 revision value.

Type: `int`. Default value: `0x1`.

ete.SIM_OVERFLOW_GRANULARITY

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

Type: `int`. Default value: `0x64`.

ete.SIM_OVERFLOW_PERCENTAGE

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

Type: `int`. Default value: `0x0`.

ete.SOURCE_ADDRESS

Allow generation of source address elements.

Type: `bool`. Default value: `0x0`.

ete.TRACE_OUTPUT

File to which to write trace byte stream.

Type: `string`. Default value: `""`.

ete.TRCRSRTA_FORCED_EXCEP

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`. Default value: `0x0`.

force_mte_tag_access_razwi_and_ignore_tag_checks

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.

Type: `bool`. Default value: `0x0`.

force_zero_mpam_partid_and_pmg

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.

Type: `int`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_enhanced_pan

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.

Type: `int`. Default value: `0x2`.

has_ete

If true, implements the Embedded Trace Extension (FEAT_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

Type: bool. Default value: 0x0.

has_large_va

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.

Type: int. Default value: 0x0.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: bool. Default value: 0x0.

has_statistical_profiling

Whether Statistical Based Profiling is implemented (FEAT_SPE).

Type: bool. Default value: 0x0.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state_modelled=true.

Type: int. Default value: 0x0.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state_modelled=true.

Type: int. Default value: 0x0.

icache-miss_latency

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.

Type: int. Default value: 0x0.

icache-prefetch_enabled

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

Type: bool. Default value: 0x0.

icache-read_access_latency

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.

Type: int. Default value: 0x0.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

instruction_tlb_size

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

Type: `int`. Default value: `0x0`.

l3cache-has_mpam

L3 Cache has MPAM support.

Type: `bool`. Default value: `0x1`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

`l3cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-snoop_issue_latency`

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`l3cache-ways`

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

Type: `int`. Default value: `0x10`.

`l3cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`l3cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`memory_tagging_support_level`

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).

Type: `int`. Default value: `0x2`.

`num_nodes`

Number of transport nodes. Zero implies direct-connect configuration.

Type: `int`. Default value: `0x1`.

`pmu-num_counters`

Number of PMU counters implemented.

Type: `int`. Default value: `0x6`.

`pseudo_fault_generation_feature_register`

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,
```

```
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.

Type: string. Default value:

```
"[{\"OF\":true,\"UC\":true,\"UEU\":false,\"UER\":false,\"UEO\":false,\"DE\":0x1,\"CE\":0x1,\"CI\":true,\"ER\":false,\"OF\":false,\"UC\":true,\"UEU\":false,\"UER\":false,\"UEO\":false,\"DE\":0x1,\"CE\":0x1,\"CI\":false,\"ER\":false}]"
```

ptw_latency

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

Type: int. Default value: 0x0.

stage12_tlb_size

Number of stage1+2 tlb entries.

Type: int. Default value: 0x80.

tlb_latency

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

Type: int. Default value: 0x0.

treat_PAC_as_NOP

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.

Type: bool. Default value: 0x0.

walk_cache_latency

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

Type: int. Default value: 0x0.

4.5.42 ARM Cortex A720CT

ARM Cortex A720CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-156: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM Cortex A720CT contains the following CADI targets:

- ARM_Cortex-A720
- Cluster_ARM_Cortex-A720
- PVCache
- TlbCadi

ARM Cortex A720CT contains the following MTI components:

- [ARM_Cortex-A720](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PPUv1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters removed:

- `advsimd_bf16_support_level`
- `fault_unalign_to_unsupported_access`
- `has_16bit_vmids`

About ARMCortexA720CT

The model supports the following features:

- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- Internal PPU support is present.
- A P-Channel for the cluster and for each core.
- `BROADCASTPERSIST` pin.
- Optional peripheral port.
- Memory-mapped register access to MPAM.
- Per-core clock.
- Utility bus.

Support for the following features is planned for a future release:

- DSU-120 system features are not fully implemented.
- Transporting architectural metadata tags over TLM2 (OSCI) and AMBA-PV (TLM2 extensions from Arm) bus protocols.
- Core-Complex.
- `BROADCASTCACHEMAINTPOU` pin
- `COREINSTRRET` and `COREINSTRRUN` signals.

- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not supported but signals are implemented.

The following features will not be implemented:

- DynamiQ™ features that are irrelevant to the programmers' view simulation, for example:
 - Automatic CPU retention mode.
 - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snooper filtering.
- 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](#).

Limitations

This model does not support the Power Control State Machine (PCSM).

Ports for ARMCortexA720CT

Table 4-157: Ports

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.

Name	Protocol	Type	Description
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[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_powerdown_out	Signal	Master	An output from PPU that informs thermal controller of the core 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_powerdown_out[14]	Signal	Master	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.

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[14]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[14]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[14]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[14]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[14]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[14]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[14]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[14]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[14]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[14]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[14]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[14]	Signal	Master	Interrupt signal from the trace buffer unit.

Name	Protocol	Type	Description
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[14]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[14]	Signal	Slave	Virtualised FIQ.
virq[14]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[14]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-A720

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.
Type: bool. Default value: 0x0.

cpu0.force-fpsid

Override the FPSID value.
Type: bool. Default value: 0x1.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.
Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.
Type: int. Default value: 0x0.

cpu0.l2cache-miss_latency

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.
Type: int. Default value: 0x0.

cpu0.l2cache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.
Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.
Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.
Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: int. Default value: 0xf000000.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: int. Default value: 0x3c.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: int. Default value: 0xab.

cpu0.trace_special_hlt_imm16

For this HLT number, IF enable_trace_special_hlt_imm16=true, skip performing usual HLT execution but call MTI trace if registered.
Type: int. Default value: 0xf000.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for Cluster_ARM_Cortex-A720**AEND0_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `int`. Default value: `0x0`.

AEND1_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `int`. Default value: `0x0`.

AEND2_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `int`. Default value: `0x0`.

AEND3_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `int`. Default value: `0x0`.

ASTART0_DEFAULT

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `int`. Default value: `0x0`.

ASTART1_DEFAULT

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `int`. Default value: `0x0`.

ASTART2_DEFAULT

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `int`. Default value: `0x0`.

ASTART3_DEFAULT

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `int`. Default value: `0x0`.

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x0`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x0`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

CPUCFR

Value of CPU Configuration Register.

Type: `int`. Default value: `0x0`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

ete.CLAIMTAGS

Number of claim tags.

Type: `int`. Default value: `0x4`.

ete.ETE_REVISION

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

Type: `int`. Default value: `0x1`.

ete.MAX_INST_PER_Q

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

Type: `int`. Default value: `0x1`.

ete.NumberOfRSPairs

Number of resource selector pairs.

Type: `int`. Default value: `0x8`.

ete.PIDR_CM0D

TRCPIDR CM0D value.

Type: `int`. Default value: `0x0`.

ete.PIDR_REVAND

TRCPIDR REVAND value.

Type: `int`. Default value: `0x0`.

`ete.PIDR_REVISION`

TRCPIDR REVISION value.

Type: `int`. Default value: `0x0`.

`ete.Q_CADENCE`

Number of instruction blocks traced between two Q elements.

Type: `int`. Default value: `0x1`.

`ete.RES0_STATEFUL`

Whether RES0 bits are stateful or RAZ/WI.

Type: `bool`. Default value: `0x0`.

`ete.RETSTACK`

Return stack depth.

Type: `int`. Default value: `0x3`.

`ete.REVISION`

TRCIDR1 revision value.

Type: `int`. Default value: `0x0`.

`ete.SIM_OVERFLOW_GRANULARITY`

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

Type: `int`. Default value: `0x64`.

`ete.SIM_OVERFLOW_PERCENTAGE`

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

Type: `int`. Default value: `0x0`.

`ete.SOURCE_ADDRESS`

Allow generation of source address elements.

Type: `bool`. Default value: `0x0`.

`ete.TRACE_OUTPUT`

File to which to write trace byte stream.

Type: `string`. Default value: `""`.

`ete.TRCSRSTA_FORCED_EXCEP`

TRCSRSTA value for a forcibly traced exception.

Type: `bool`. Default value: `0x0`.

`ete.TSMARK`

Whether timestamp markers are supported.

Type: `bool`. Default value: `0x1`.

`force_mte_tag_access_razwi_and_ignore_tag_checks`

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.

Type: `bool`. Default value: `0x0`.

force_zero_mpam_partid_and_pmg

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.

Type: `int`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_enhanced_pan

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.

Type: `int`. Default value: `0x2`.

has_ete

If true, implements the Embedded Trace Extension (FEAT_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

Type: `bool`. Default value: `0x0`.

has_large_va

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.

Type: `int`. Default value: `0x0`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

has_statistical_profiling

Whether Statistical Based Profiling is implemented (FEAT_SPE).

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

icache-read_latency

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.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

instruction_tlb_size

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

Type: `int`. Default value: `0x0`.

l3cache-has_mpam

L3 Cache has MPAM support.

Type: `bool`. Default value: `0x1`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-ways

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

Type: `int`. Default value: `0x10`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

memory_tagging_support_level

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).

Type: `int`. Default value: `0x3`.

num_acp

Number of ACP ports.

Type: `int`. Default value: `0x0`.

NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

num_nodes

Number of transport nodes. Zero implies direct-connect configuration.

Type: `int`. Default value: `0x1`.

pmu-num_counters

Number of PMU counters implemented.

Type: `int`. Default value: `0x6`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

stage12_tlb_size

Number of stage1+2 tlb entries.

Type: `int`. Default value: `0x80`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

treat_PAC_as_NOP

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.

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.43 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 4-158: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexM0CT contains the following CADI targets:

- ARM_Cortex-M0

ARMCortexM0CT contains the following MTI components:

- [ARM_Cortex-M0](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)

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 through CADI a VTOR register, but this register does not exist in the IP.
- Arm®v6-M is a subset of Arm®v7-M. Arm does not guarantee that all Arm®v7-M-specific behavior is absent from Arm®v6-M Fast Models cores. Therefore, Arm does not guarantee that code that runs on Arm®v7-M cores but fails on Arm®v6-M cores will also fail on Arm®v6-M Fast Models cores.

Ports for ARMCortexM0CT

Table 4-159: Ports

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
currpri	Value	Master	Current execution priority.
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr[32]	Signal	Slave	This signal array delivers signals to the NVIC.
intnmi	Signal	Slave	Configure non maskable interrupt.
lockup	Signal	Master	Asserted when the processor is in lockup state.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.

Name	Protocol	Type	Description
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-M0

BIGENDINIT

Initialize processor to big endian mode.
Type: `bool`. Default value: `0x0`.

BKPT

Number of breakpoint unit comparators implemented.
Type: `int`. Default value: `0x4`.

cpi_div

divider for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

cpi_mul

multiplier for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

DBG

Set whether debug extensions are implemented.
Type: `bool`. Default value: `0x1`.

master_id

Master ID presented in bus transactions.
Type: `int`. Default value: `0x0`.

min_sync_level

force minimum syncLevel (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll).
Type: `int`. Default value: `0x0`.

NUM_IRQ

Number of user interrupts.
Type: `int`. Default value: `0x20`.

semihosting-cmd_line

Command line available to semihosting SVC calls.
Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.
Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x20700000`.

semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x20800000`.

semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0x20700000`.

semihosting-Thumb_SVC

T32 SVC number for semihosting.

Type: `int`. Default value: `0xab`.

SYST

Enable support for SysTick timer functionality.

Type: `bool`. Default value: `0x1`.

use_architectural_names

Enable using architectural names for registers LR/PSP/MSP instead of model-specific names.

Type: `bool`. Default value: `0x0`.

WIC

Include support for WIC-mode deep sleep.

Type: `bool`. Default value: `0x1`.

WPT

Number of watchpoint unit comparators implemented.

Type: `int`. Default value: `0x2`.

4.5.44 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 4-160: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexM0PlusCT contains the following CADI targets:

- ARM_Cortex-M0+

ARMCortexM0PlusCT contains the following MTI components:

- [ARM_Cortex-M0plus](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)

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.
- Arm®v6-M is a subset of Arm®v7-M. Arm does not guarantee that all Arm®v7-M-specific behavior is absent from Arm®v6-M Fast Models cores. Therefore, Arm does not guarantee that code that runs on Arm®v7-M cores but fails on Arm®v6-M cores will also fail on Arm®v6-M Fast Models cores.

Ports for ARMCortexM0PlusCT

Table 4-161: Ports

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cpuwait	Signal	Slave	CPUWAIT extends effect of reset when true
currpri	Value	Master	Current execution priority.
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr[32]	Signal	Slave	This signal array delivers signals to the NVIC.
intnmi	Signal	Slave	Configure non maskable interrupt.
io_port_in	PVBus	Slave	I/O port pair. See the documentation for the io_port_out port.

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.
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_Cortex-M0plus

BIGENDINIT

Initialize processor to big endian mode.
Type: `bool`. Default value: `0x0`.

BKPT

Number of breakpoint unit comparators implemented.
Type: `int`. Default value: `0x4`.

cpi_div

divider for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

cpi_mul

multiplier for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

DBG

Set whether debug extensions are implemented.
Type: `bool`. Default value: `0x1`.

IOP

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.
Type: `bool`. Default value: `0x0`.

IRQDIS

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n].

Type: `int`. Default value: `0x0`.

master_id

Master ID presented in bus transactions.

Type: `int`. Default value: `0x0`.

min_sync_level

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

Type: `int`. Default value: `0x0`.

NUM_IRQ

Number of user interrupts.

Type: `int`. Default value: `0x20`.

NUM_MPU_REGION

Number of MPU regions.

Type: `int`. Default value: `0x0`.

scheduler_mode

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, 3=ISSCompare) WARNING:

This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `int`. Default value: `0x0`.

semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x20700000`.

semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

- semihosting-stack_base**
Virtual address of base of descending stack.
Type: `int`. Default value: `0x20800000`.
- semihosting-stack_limit**
Virtual address of stack limit.
Type: `int`. Default value: `0x20700000`.
- semihosting-Thumb_SVC**
T32 SVC number for semihosting.
Type: `int`. Default value: `0xab`.
- SYST**
Enable support for SysTick timer functionality.
Type: `bool`. Default value: `0x1`.
- use_architectural_names**
Enable using architectural names for registers LR/PSP/MSP instead of model-specific names.
Type: `bool`. Default value: `0x0`.
- USER**
Enable support for Unprivileged/Privileged Extension.
Type: `bool`. Default value: `0x0`.
- VTOR**
Include Vector Table Offset Register.
Type: `bool`. Default value: `0x0`.
- WIC**
Include support for WIC-mode deep sleep.
Type: `bool`. Default value: `0x1`.
- WPT**
Number of watchpoint unit comparators implemented.
Type: `int`. Default value: `0x2`.

4.5.45 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 4-162: IP revisions support

Revision	Quality level
r2p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexM3CT contains the following CADI targets:

- ARM_Cortex-M3

ARMCortexM3CT contains the following MTI components:

- [ARM_Cortex-M3](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)

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 `p_vbus_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 `p_v_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:

Table 4-163: 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 ARMCortexM3CT

Table 4-164: Ports

Name	Protocol	Type	Description
ahb_ap	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
auxfault	Value	Slave	This is wired to the Auxiliary Fault Status Register.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
currpri	Value	Master	Current execution priority.
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr[240]	Signal	Slave	This signal array delivers signals to the NVIC.
intnmi	Signal	Slave	Configure non maskable interrupt.
lockup	Signal	Master	Asserted when the processor is in lockup state.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
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_Cortex-M3

BB_PRESENT

Enable bitbanding.

Type: `bool`. Default value: `0x1`.

BIGENDINIT

Initialize processor to big endian mode.

Type: `bool`. Default value: `0x0`.

cpi_div

divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

DBG_LVL

0: No debug; 1: Minimal debug; 2: Full debug without DWT address matching; 3: Full debug support with DWT data-comparators.

Type: `int`. Default value: `0x3`.

LVL_WIDTH

Number of bits of interrupt priority.

Type: `int`. Default value: `0x3`.

master_id

Master ID presented in bus transactions.

Type: `int`. Default value: `0x0`.

min_sync_level

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

Type: `int`. Default value: `0x0`.

NUM_IRQ

Number of user interrupts.

Type: `int`. Default value: `0x10`.

NUM_MPU_REGION

Number of MPU regions.

Type: `int`. Default value: `0x8`.

scheduler_mode

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, 3=ISSCompare) WARNING:

This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `int`. Default value: `0x0`.

semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x20700000`.

semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x20800000`.

semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0x20700000`.

semihosting-Thumb_SVC

T32 SVC number for semihosting.

Type: `int`. Default value: `0xab`.

TRACE_LVL

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.

Type: `int`. Default value: `0x1`.

use_architectural_names

Enable using architectural names for registers LR/PSP/MSP instead of model-specific names.

Type: `bool`. Default value: `0x0`.

WIC

Include support for WIC-mode deep sleep.

Type: `bool`. Default value: `0x1`.

4.5.46 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 4-165: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexM4CT contains the following CADI targets:

- ARM_Cortex-M4

ARMCortexM4CT contains the following MTI components:

- [ARM_Cortex-M4](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)

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 `p_vbus_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 `p_v_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 `dbgcn` 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:

Table 4-166: 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 ARMCortexM4CT

Table 4-167: 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 <code>clk_in</code> port is used to determine the rate at which the core executes instructions.
currpri	Value	Master	Current execution priority.
dbgcn	Signal	Slave	Disallow (DAP) debugger access.
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpudisable	Signal	Slave	Configure core with no FPU on reset.

Name	Protocol	Type	Description
fpxxc	Value	Master	Port that sends the value of the FPxxC exception flags (FPIX, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
intisr[240]	Signal	Slave	This signal array delivers signals to the NVIC.
intnmi	Signal	Slave	Configure non maskable interrupt.
lockup	Signal	Master	Asserted when the processor is in lockup state.
mpudisable	Signal	Slave	Configure core with no MPU on reset.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
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-M4

BB_PRESENT

Enable bitbanding.

Type: `bool`. Default value: `0x1`.

BIGENDINIT

Initialize processor to big endian mode.

Type: `bool`. Default value: `0x0`.

cpi_div

divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

DBGLVL

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.

Type: `int`. Default value: `0x3`.

LVL_WIDTH

Number of bits of interrupt priority.

Type: `int`. Default value: `0x3`.

master_id

Master ID presented in bus transactions.

Type: `int`. Default value: `0x0`.

min_sync_level

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

Type: `int`. Default value: `0x0`.

NUM_IRQ

Number of user interrupts.

Type: `int`. Default value: `0x10`.

NUM_MPU_REGION

Number of MPU regions.

Type: `int`. Default value: `0x8`.

scheduler_mode

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, 3=ISSCompare) WARNING: This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `int`. Default value: `0x0`.

semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x20700000`.

semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x20800000`.

semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0x20700000`.

semihosting-Thumb_SVC

T32 SVC number for semihosting.
Type: `int`. Default value: `0xab`.

TRACE_LVL

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.
Type: `int`. Default value: `0x1`.

use_architectural_names

Enable using architectural names for registers LR/PSP/MSP instead of model-specific names.
Type: `bool`. Default value: `0x0`.

vfp-present

Set whether the model has VFP support.
Type: `bool`. Default value: `0x1`.

WIC

Include support for WIC-mode deep sleep.
Type: `bool`. Default value: `0x1`.

4.5.47 **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 4-168: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexM7CT contains the following CADI targets:

- `ARM_Cortex-M7`

ARMCortexM7CT contains the following MTI components:

- [ARM_Cortex-M7](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)

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:

Table 4-169: 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 ARMCortexM7CT

Table 4-170: 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 <code>clk_in</code> 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.

Name	Protocol	Type	Description
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_Cortex-M7

BIGENDINIT

Initialize processor to big endian mode.

Type: bool. Default value: 0x0.

CFGAHBPSZ

Size of the AHBP port memory region. 0=AHBP disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

Type: int. Default value: 0x0.

cpi_div

divider for calculating CPI (Cycles Per Instruction).

Type: int. Default value: 0x1.

cpi_mul

multiplier for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

CTI

CTI (Cross Trigger Interface) included.
Type: `bool`. Default value: `0x0`.

CTI_irq0_pin

CTI interrupt request 0 pin.
Type: `int`. Default value: `0x4`.

CTI_irq1_pin

CTI interrupt request 1 pin.
Type: `int`. Default value: `0x5`.

DBGLVL

0: 2 DWT, 4 FPB; 1: 4 DWT, 8 FPB comparators.
Type: `int`. Default value: `0x1`.

dcache-size

L1 D-cache size in bytes.
Type: `int`. Default value: `0x8000`.

dcache-state_modelled

Set whether D-cache has stateful implementation.
Type: `bool`. Default value: `0x0`.

dcache-ways

L1 D-cache ways (sets are implicit from size).
Type: `int`. Default value: `0x4`.

DP_FLOAT

Support 8-byte floats.
Type: `bool`. Default value: `0x1`.

dtcm_enable

Enable DTCM at reset.
Type: `bool`. Default value: `0x0`.

dtcm_size

DTCM size in KB.
Type: `int`. Default value: `0x100`.

duplicate_CADI_TCM_writes

CADI writes to TCMs are also sent to downstream memory at same addresses (for validation platforms).
Type: `bool`. Default value: `0x0`.

icache-size

L1 I-cache size in bytes.
Type: `int`. Default value: `0x8000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

icache-ways

L1 I-cache ways (sets are implicit from size).

Type: `int`. Default value: `0x2`.

INITAHBPEN

The AHBP enable state at reset.

Type: `bool`. Default value: `0x0`.

INITVTOR

vector-table offset at reset.

Type: `int`. Default value: `0x0`.

itcm_enable

Enable ITCM at reset.

Type: `bool`. Default value: `0x0`.

itcm_size

ITCM size in KB.

Type: `int`. Default value: `0x100`.

LVL_WIDTH

Number of bits of interrupt priority.

Type: `int`. Default value: `0x3`.

master_id

Master ID presented in bus transactions.

Type: `int`. Default value: `0x0`.

min_sync_level

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

Type: `int`. Default value: `0x0`.

NUM_IRQ

Number of user interrupts.

Type: `int`. Default value: `0x20`.

NUM_MPU_REGION

Number of MPU regions.

Type: `int`. Default value: `0x10`.

scheduler_mode

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, 3=ISSCompare) WARNING: This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `int`. Default value: `0x0`.

semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x20700000`.

semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x20800000`.

semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0x20700000`.

semihosting-Thumb_SVC

T32 SVC number for semihosting.

Type: `int`. Default value: `0xab`.

TRC

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

Type: `bool`. Default value: `0x1`.

use_architectural_names

Enable using architectural names for registers LR/PSP/MSP instead of model-specific names.

Type: `bool`. Default value: `0x0`.

vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

WIC

Include support for WIC-mode deep sleep.

Type: `bool`. Default value: `0x1`.

4.5.48 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 4-171: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexM23CT contains the following CADI targets:

- ARM_Cortex-M23

ARMCortexM23CT contains the following MTI components:

- [ARM_Cortex-M23](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)

Changes in 11.24.4

Ports added:

- wicenack
- wicenreq
- wicsense

Parameters added:

- WICLINES

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 4-172: 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	Clear = Core goes through reset sequence as normal, Set = Core waits out of reset.

Name	Protocol	Type	Description
currpri	Value	Master	Current execution priority.
dap_s	PVBus	Slave	Debug Access Port (DAP).
dbgen	Signal	Slave	Invasive debug control signals. Debug enable, Set=enabled, Clear=disabled
dbgrestart	Signal	Slave	External request to leave debug state
dbgrestarted	Signal	Master	Acknowledge for DBGRESTART
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
halted	Signal	Master	Core is in halt mode debug state
hreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
idau	PVBus	Master	The core will generate IDAU requests on this port.
idau_invalidate_region	Value_64	Slave	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
initvtor	Value	Slave	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initvtorns	Value	Slave	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
io_port_in	PVBus	Slave	I/O port pair. See the documentation for the io_port_out port.
io_port_out	PVBus	Master	I/O port pair. Used if IOP is true. Transactions from io_port_out which do not "match" should be returned via io_port_in. For performance reasons, the I/O port interface is not modelled directly. Instead, a simple PVBus gasket is inserted at the point in the memory system where the I/O port would be. In hardware, a device would be attached to the port which would tell the CPU whether it would like to intercept each transaction, given its address. This can be modelled in a performant manner by connecting a PVBusMapper-based device to io_port_out which intercepts transactions of interest and passes other transactions back to the CPU via io_port_in. Your I/O port device model is also responsible for aborting transactions which would be aborted on hardware (e.g. exclusives) if necessary.
irq[240]	Signal	Slave	This signal array delivers signals to the NVIC.
lockup	Signal	Master	Asserted when the processor is in lockup state.
niden	Signal	Slave	Non-invasive debug enable, Set=enabled, Clear=disabled
nmi	Signal	Slave	Configure non maskable interrupt.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
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

Name	Protocol	Type	Description
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 ARM_Cortex-M23

BE

Initialize processor to big endian mode.
Type: `bool`. Default value: `0x0`.

BKPT

Number of breakpoint unit comparators implemented.
Type: `int`. Default value: `0x4`.

cpi_div

divider for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

cpi_mul

multiplier for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

CTI

CTI (Cross Trigger Interface) included.
Type: `bool`. Default value: `0x0`.

CTI_irq0_pin

CTI interrupt request 0 pin.
Type: `int`. Default value: `0x4`.

CTI_irq1_pin

CTI interrupt request 1 pin.
Type: `int`. Default value: `0x5`.

DBG

Set whether debug extensions are implemented.
Type: `bool`. Default value: `0x1`.

ignore-SCR.SLEEPONEXIT

Never sleep on exit from handler to thread mode.
Type: `bool`. Default value: `0x0`.

INITVTOR

Secure vector-table offset at reset.

Type: `int`. Default value: `0x0`.

INITVTORNS

Non-Secure vector-table offset at reset.

Type: `int`. Default value: `0x0`.

IOP

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`.

Type: `bool`. Default value: `0x0`.

IRQDIS0

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+0]`.

Type: `int`. Default value: `0x0`.

IRQDIS1

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+32]`.

Type: `int`. Default value: `0x0`.

IRQDIS2

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+64]`.

Type: `int`. Default value: `0x0`.

IRQDIS3

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+96]`.

Type: `int`. Default value: `0x0`.

IRQDIS4

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+128]`.

Type: `int`. Default value: `0x0`.

IRQDIS5

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+160]`.

Type: `int`. Default value: `0x0`.

IRQDIS6

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+192]`.

Type: `int`. Default value: `0x0`.

IRQDIS7

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+224]`.

Type: `int`. Default value: `0x0`.

master_id

Master ID presented in bus transactions.

Type: `int`. Default value: `0x0`.

min_sync_level

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

Type: `int`. Default value: `0x0`.

MPU_NS

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

Type: `int`. Default value: `0x8`.

MPU_S

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

Type: `int`. Default value: `0x8`.

NUMIRQ

Number of user interrupts.

Type: `int`. Default value: `0x10`.

SAU

Number of SAU regions (0 => no SAU).

Type: `int`. Default value: `0x4`.

SAU_CTRL.ALLNS

At reset, the SAU treats entire memory space as NS when the SAU is disabled if this is set.

Type: `bool`. Default value: `0x0`.

SAU_CTRL.ENABLE

Enable SAU at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION0.BADDR

Base address of SAU region0 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION0.ENABLE

Enable SAU region0 at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION0.LADDR

Limit address of SAU region0 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION0.NSC

Set NSC for SAU region0 at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION1.BADDR

Base address of SAU region1 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION1.ENABLE

Enable SAU region1 at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION1.LADDR

Limit address of SAU region1 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION1.NSC

Set NSC for SAU region1 at reset.
Type: bool. Default value: 0x0.

SAU_REGION2.BADDR

Base address of SAU region2 at reset.
Type: int. Default value: 0x0.

SAU_REGION2.ENABLE

Enable SAU region2 at reset.
Type: bool. Default value: 0x0.

SAU_REGION2.LADDR

Limit address of SAU region2 at reset.
Type: int. Default value: 0x0.

SAU_REGION2.NSC

Set NSC for SAU region2 at reset.
Type: bool. Default value: 0x0.

SAU_REGION3.BADDR

Base address of SAU region3 at reset.
Type: int. Default value: 0x0.

SAU_REGION3.ENABLE

Enable SAU region3 at reset.
Type: bool. Default value: 0x0.

SAU_REGION3.LADDR

Limit address of SAU region3 at reset.
Type: int. Default value: 0x0.

SAU_REGION3.NSC

Set NSC for SAU region3 at reset.
Type: bool. Default value: 0x0.

SAU_REGION4.BADDR

Base address of SAU region4 at reset.
Type: int. Default value: 0x0.

SAU_REGION4.ENABLE

Enable SAU region4 at reset.
Type: bool. Default value: 0x0.

SAU_REGION4.LADDR

Limit address of SAU region4 at reset.
Type: int. Default value: 0x0.

SAU_REGION4.NSC

Set NSC for SAU region4 at reset.
Type: bool. Default value: 0x0.

SAU_REGION5.BADDR

Base address of SAU region5 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION5.ENABLE

Enable SAU region5 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION5.LADDR

Limit address of SAU region5 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION5.NSC

Set NSC for SAU region5 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION6.BADDR

Base address of SAU region6 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION6.ENABLE

Enable SAU region6 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION6.LADDR

Limit address of SAU region6 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION6.NSC

Set NSC for SAU region6 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION7.BADDR

Base address of SAU region7 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION7.ENABLE

Enable SAU region7 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION7.LADDR

Limit address of SAU region7 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION7.NSC

Set NSC for SAU region7 at reset.
Type: `bool`. Default value: `0x0`.

scheduler_mode

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, 3=ISSCompare) WARNING:

This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `int`. Default value: `0x0`.

SECEXT

Whether the ARMv8-M Security Extensions are included.

Type: `bool`. Default value: `0x1`.

semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x20700000`.

semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x20800000`.

semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0x20700000`.

semihosting-Thumb_SVC

T32 SVC number for semihosting.

Type: `int`. Default value: `0xab`.

SYST

Include SysTick timer functionality (0=Absent, 1=Secure only, 2=Secure and NS).

Type: `int`. Default value: `0x2`.

use_architectural_names

Enable using architectural names for registers LR/PSP/MSP instead of model-specific names.

Type: `bool`. Default value: `0x0`.

VTOR

Include Vector Table Offset Register.
Type: `bool`. Default value: `0x1`.

WIC

Include support for WIC-mode deep sleep.
Type: `bool`. Default value: `0x1`.

WICLINES

Number of lines supported by the WIC interface.
Type: `int`. Default value: `0x12`.

WPT

Number of watchpoint unit comparators implemented.
Type: `int`. Default value: `0x4`.

4.5.49 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 4-173: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexM33CT contains the following CADI targets:

- `ARM_Cortex-M33`

ARMCortexM33CT contains the following MTI components:

- [ARM_Cortex-M33](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)

About ARMCortexM33CT

The model supports Custom Datapath Extension (CDE). For more information, see [5.7 CDE](#) on page 1628.

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:

Table 4-174: 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 ARMCortexM33CT

Table 4-175: 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 <code>clk_in</code> 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.

Name	Protocol	Type	Description
fpxxc	Value	Master	Port that sends the value of the FPxxC exception flags (FPIXC, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	Signal	Master	Indicates that the processor is in halt mode
idau	PVBus	Master	The core will generate IDAU requests on this port.
idau_invalidate_region	Value_64	Slave	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
initnsvtor	Value	Slave	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initsvtor	Value	Slave	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
intnum	Value	Master	Exception number of the current execution context (from IPSR[8:0]) When the processor is in Thread mode, INTNUM is 0 When the processor is in Handler mode, INTNUM is the exception number of the currently-executing exception.
irq[480]	Signal	Slave	This signal array delivers signals to the NVIC.
locknsmpu	Signal	Slave	Disable writes to the Non-Secure MPU_*_NS registers
locknsvtor	Signal	Slave	Cortex-M33-specific LOCKNSVTOR, LOCKSVTAIRCR, LOCKSMPU, LOCKNSMPU, LOCKSAU. Disable writes to VTOR_NS
locksau	Signal	Slave	Disable writes to the SAU_* registers
locksmpu	Signal	Slave	Disable writes to the Secure MPU_* registers
locksvtaircr	Signal	Slave	Disable writes to VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS
lockup	Signal	Master	Asserted when the processor is in lockup state.
niden	Signal	Slave	Non-invasive debug enable
nmi	Signal	Slave	Configure non maskable interrupt.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pdbus_m	PVBus	Master	The core will generate bus requests on this port.
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
spiden	Signal	Slave	Secure invasive debug enable
spniden	Signal	Slave	Secure non-invasive debug enable
stcalib[2]	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.
wicenack	Signal	Master	Acknowledge signal for WICENREQ

Name	Protocol	Type	Description
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_Cortex-M33

BIGENDINIT

Initialize processor to big endian mode.

Type: `bool`. Default value: `0x0`.

cde_impl_name

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

Type: `string`. Default value: `""`.

CDEMAPPEDONCP

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

Type: `int`. Default value: `0xff`.

CDERTLID

Value of ID_AFR0.CDERTLID.

Type: `int`. Default value: `0x20`.

CFGNOCDECP

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

Type: `int`. Default value: `0x0`.

cpi_div

divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

CPIF

Specifies whether the external coprocessor interface is included.

Type: `bool`. Default value: `0x1`.

CPNSPRESENT

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

Type: `int`. Default value: `0xff`.

CPSPRESENT

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

Type: `int`. Default value: `0xff`.

CTI

CTI (Cross Trigger Interface) included.

Type: `bool`. Default value: `0x0`.

CTI_irq0_pin

CTI interrupt request 0 pin.
Type: `int`. Default value: `0x4`.

CTI_irq1_pin

CTI interrupt request 1 pin.
Type: `int`. Default value: `0x5`.

DBGLVL

0: Minimal debug; 1: 2 Watchpoints, 4 Breakpoint comparators; 2: 4 Watchpoints, 8 Breakpoint comparators.
Type: `int`. Default value: `0x2`.

DSP

Set whether the model has the DSP extension.
Type: `bool`. Default value: `0x1`.

FPU

Set whether the model has VFP support.
Type: `bool`. Default value: `0x1`.

has_cde

Enables Custom Datapath Extensions.
Type: `bool`. Default value: `0x0`.

ignore-SCR.SLEEPONEXIT

Never sleep on exit from handler to thread mode.
Type: `bool`. Default value: `0x0`.

INITNSVTOR

Non-Secure vector-table offset at reset.
Type: `int`. Default value: `0x0`.

INITSVTOR

Secure vector-table offset at reset.
Type: `int`. Default value: `0x0`.

IRQDIS0

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+0].
Type: `int`. Default value: `0x0`.

IRQDIS1

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+32].
Type: `int`. Default value: `0x0`.

IRQDIS10

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+320].
Type: `int`. Default value: `0x0`.

IRQDIS11

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+352].
Type: `int`. Default value: `0x0`.

IRQDIS12

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

Type: `int`. Default value: `0x0`.

IRQDIS13

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

Type: `int`. Default value: `0x0`.

IRQDIS14

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

Type: `int`. Default value: `0x0`.

IRQDIS2

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

Type: `int`. Default value: `0x0`.

IRQDIS3

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

Type: `int`. Default value: `0x0`.

IRQDIS4

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

Type: `int`. Default value: `0x0`.

IRQDIS5

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

Type: `int`. Default value: `0x0`.

IRQDIS6

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

Type: `int`. Default value: `0x0`.

IRQDIS7

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

Type: `int`. Default value: `0x0`.

IRQDIS8

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

Type: `int`. Default value: `0x0`.

IRQDIS9

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

Type: `int`. Default value: `0x0`.

IRQLVL

Number of bits of interrupt priority.

Type: `int`. Default value: `0x3`.

ITM

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

Type: `bool`. Default value: `0x1`.

LOCK_NS_MPU

Lock down of Non-Secure MPU registers write.

Type: `bool`. Default value: `0x0`.

LOCK_S_MPU

Lock down of Secure MPU registers write.

Type: `bool`. Default value: `0x0`.

LOCK_SAU

Lock down of SAU registers write.

Type: `bool`. Default value: `0x0`.

master_id

Master ID presented in bus transactions.

Type: `int`. Default value: `0x0`.

min_sync_level

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

Type: `int`. Default value: `0x0`.

MPU_NS

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

Type: `int`. Default value: `0x8`.

MPU_S

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

Type: `int`. Default value: `0x8`.

NUMIRQ

Number of user interrupts.

Type: `int`. Default value: `0x20`.

SAU

Number of SAU regions (0 => no SAU).

Type: `int`. Default value: `0x4`.

SAU_CTRL.ALLNS

At reset, the SAU treats entire memory space as NS when the SAU is disabled if this is set.

Type: `bool`. Default value: `0x0`.

SAU_CTRL.ENABLE

Enable SAU at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION0.BADDR

Base address of SAU region0 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION0.ENABLE

Enable SAU region0 at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION0.LADDR

Limit address of SAU region0 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION0.NSC

Set NSC for SAU region0 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION1.BADDR

Base address of SAU region1 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION1.ENABLE

Enable SAU region1 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION1.LADDR

Limit address of SAU region1 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION1.NSC

Set NSC for SAU region1 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION2.BADDR

Base address of SAU region2 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION2.ENABLE

Enable SAU region2 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION2.LADDR

Limit address of SAU region2 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION2.NSC

Set NSC for SAU region2 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION3.BADDR

Base address of SAU region3 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION3.ENABLE

Enable SAU region3 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION3.LADDR

Limit address of SAU region3 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION3.NSC

Set NSC for SAU region3 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION4.BADDR

Base address of SAU region4 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION4.ENABLE

Enable SAU region4 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION4.LADDR

Limit address of SAU region4 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION4.NSC

Set NSC for SAU region4 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION5.BADDR

Base address of SAU region5 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION5.ENABLE

Enable SAU region5 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION5.LADDR

Limit address of SAU region5 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION5.NSC

Set NSC for SAU region5 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION6.BADDR

Base address of SAU region6 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION6.ENABLE

Enable SAU region6 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION6.LADDR

Limit address of SAU region6 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION6.NSC

Set NSC for SAU region6 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION7.BADDR

Base address of SAU region7 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION7.ENABLE

Enable SAU region7 at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION7.LADDR

Limit address of SAU region7 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION7.NSC

Set NSC for SAU region7 at reset.

Type: `bool`. Default value: `0x0`.

scheduler_mode

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, 3=ISSCompare) WARNING: This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `int`. Default value: `0x0`.

SECEXT

Whether the ARMv8-M Security Extensions are included.

Type: `bool`. Default value: `0x1`.

semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x20700000`.

semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

semihosting-stack_base
Virtual address of base of descending stack.
Type: `int`. Default value: `0x20800000`.

semihosting-stack_limit
Virtual address of stack limit.
Type: `int`. Default value: `0x20700000`.

semihosting-Thumb_SVC
T32 SVC number for semihosting.
Type: `int`. Default value: `0xab`.

use_architectural_names
Enable using architectural names for registers LR/PSP/MSP instead of model-specific names.
Type: `bool`. Default value: `0x0`.

vfp-enable_at_reset
Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: `bool`. Default value: `0x0`.

WIC
Include support for WIC-mode deep sleep.
Type: `bool`. Default value: `0x1`.

WICLINES
Number of lines supported by the WIC interface.
Type: `int`. Default value: `0x23`.

write_unknown_regs_at_exception
DEPRECATED and non-functional in device models.
Type: `bool`. Default value: `0x0`.

4.5.50 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 4-176: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexM35PCT contains the following CADI targets:

- ARM_Cortex-M35P

ARMCortexM35PCT contains the following MTI components:

- [ARM_Cortex-M35P](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)

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:

Table 4-177: ITM_PACKET_TYPE field values that the model supports

Field value	Description	Supported by model
<code>ITM_SYNC</code>	Synchronization packet	Not supported.
<code>ITM_P_OVERFLOW</code>	Protocol: Overflow packet	Not supported.
<code>ITM_P_LOCAL_TIMESTAMP</code>	Protocol: Local timestamp packets	Not supported.
<code>ITM_P_GLOBAL_TIMESTAMP</code>	Protocol: Global timestamp packets	Not supported.
<code>ITM_P_EXTEN</code>	Protocol: Extension packet	Not supported.
<code>ITM_S_INSTRUMENTATION</code>	Source: Instrumentation packet	Supported.
<code>ITM_S_DWT_EVENT_COUNTER</code>	Hardware source: Event counter wrapping	Not supported.
<code>ITM_S_DWT_EXCEPTION</code>	Hardware source: Exception tracing	Supported.
<code>ITM_S_DWT_PC_SAMPLING</code>	Hardware source: PC sampling	Not supported.
<code>ITM_S_DWT_DATA_PC_TRACE</code>	Hardware source: DWT Data trace PC value	Supported.
<code>ITM_S_DWT_DATA_ADDRESS_TRACE</code>	Hardware source: DWT Data trace address value	Supported.
<code>ITM_S_DWT_DATA_DATA_TRACE</code>	Hardware source: DWT Data trace DATA value	Supported.

Ports for ARMCortexM35PCT

Table 4-178: 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.
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.

Name	Protocol	Type	Description
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.
wicsense[483]	Signal	Master	Indicates which input events can cause the WIC to generate the WAKEUP signal.

Parameters for ARM_Cortex-M35P

ATFINITEN

ATFCR is enabled when the core goes out of reset.
Type: bool. Default value: 0x0.

BIGENDINIT

Initialize processor to big endian mode.
Type: bool. Default value: 0x0.

cpi_div

divider for calculating CPI (Cycles Per Instruction).
Type: int. Default value: 0x1.

cpi_mul

multiplier for calculating CPI (Cycles Per Instruction).
Type: int. Default value: 0x1.

CPIF

Specifies whether the external coprocessor interface is included.
Type: bool. Default value: 0x1.

CPNSPRESENT

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.
Type: int. Default value: 0xff.

CPSPRESENT

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

Type: `int`. Default value: `0xff`.

CTI

CTI (Cross Trigger Interface) included.

Type: `bool`. Default value: `0x0`.

CTI_irq0_pin

CTI interrupt request 0 pin.

Type: `int`. Default value: `0x4`.

CTI_irq1_pin

CTI interrupt request 1 pin.

Type: `int`. Default value: `0x5`.

DBG_LVL

0: Minimal debug; 1: 2 Watchpoints, 4 Breakpoint comparators; 2: 4 Watchpoints, 8 Breakpoint comparators.

Type: `int`. Default value: `0x2`.

DSP

Set whether the model has the DSP extension.

Type: `bool`. Default value: `0x1`.

FPU

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

ignore-SCR.SLEEPONEXIT

Never sleep on exit from handler to thread mode.

Type: `bool`. Default value: `0x0`.

INITNSVTOR

Non-Secure vector-table offset at reset.

Type: `int`. Default value: `0x0`.

INITSVTOR

Secure vector-table offset at reset.

Type: `int`. Default value: `0x0`.

IRQDIS0

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

Type: `int`. Default value: `0x0`.

IRQDIS1

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

Type: `int`. Default value: `0x0`.

IRQDIS10

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

Type: `int`. Default value: `0x0`.

IRQDIS11

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].
Type: `int`. Default value: `0x0`.

IRQDIS12

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].
Type: `int`. Default value: `0x0`.

IRQDIS13

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].
Type: `int`. Default value: `0x0`.

IRQDIS14

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].
Type: `int`. Default value: `0x0`.

IRQDIS2

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].
Type: `int`. Default value: `0x0`.

IRQDIS3

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].
Type: `int`. Default value: `0x0`.

IRQDIS4

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].
Type: `int`. Default value: `0x0`.

IRQDIS5

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].
Type: `int`. Default value: `0x0`.

IRQDIS6

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].
Type: `int`. Default value: `0x0`.

IRQDIS7

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].
Type: `int`. Default value: `0x0`.

IRQDIS8

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].
Type: `int`. Default value: `0x0`.

IRQDIS9

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].
Type: `int`. Default value: `0x0`.

IRQLVL

Number of bits of interrupt priority.
Type: `int`. Default value: `0x3`.

ITM

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

Type: bool. Default value: 0x1.

LOCK_NS_MPU

Lock down of Non-Secure MPU registers write.

Type: bool. Default value: 0x0.

LOCK_S_MPU

Lock down of Secure MPU registers write.

Type: bool. Default value: 0x0.

LOCK_SAU

Lock down of SAU registers write.

Type: bool. Default value: 0x0.

master_id

Master ID presented in bus transactions.

Type: int. Default value: 0x0.

min_sync_level

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

Type: int. Default value: 0x0.

MPU_NS

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

Type: int. Default value: 0x8.

MPU_S

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

Type: int. Default value: 0x8.

NUMIRQ

Number of user interrupts.

Type: int. Default value: 0x20.

SAU

Number of SAU regions (0 => no SAU).

Type: int. Default value: 0x4.

SAU_CTRL.ALLNS

At reset, the SAU treats entire memory space as NS when the SAU is disabled if this is set.

Type: bool. Default value: 0x0.

SAU_CTRL.ENABLE

Enable SAU at reset.

Type: bool. Default value: 0x0.

SAU_REGION0.BADDR

Base address of SAU region0 at reset.

Type: int. Default value: 0x0.

SAU_REGION0.ENABLE

Enable SAU region0 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION0.LADDR

Limit address of SAU region0 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION0.NSC

Set NSC for SAU region0 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION1.BADDR

Base address of SAU region1 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION1.ENABLE

Enable SAU region1 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION1.LADDR

Limit address of SAU region1 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION1.NSC

Set NSC for SAU region1 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION2.BADDR

Base address of SAU region2 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION2.ENABLE

Enable SAU region2 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION2.LADDR

Limit address of SAU region2 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION2.NSC

Set NSC for SAU region2 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION3.BADDR

Base address of SAU region3 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION3.ENABLE

Enable SAU region3 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION3.LADDR

Limit address of SAU region3 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION3.NSC

Set NSC for SAU region3 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION4.BADDR

Base address of SAU region4 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION4.ENABLE

Enable SAU region4 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION4.LADDR

Limit address of SAU region4 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION4.NSC

Set NSC for SAU region4 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION5.BADDR

Base address of SAU region5 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION5.ENABLE

Enable SAU region5 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION5.LADDR

Limit address of SAU region5 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION5.NSC

Set NSC for SAU region5 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION6.BADDR

Base address of SAU region6 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION6.ENABLE

Enable SAU region6 at reset.
Type: `bool`. Default value: `0x0`.

SAU_REGION6.LADDR

Limit address of SAU region6 at reset.
Type: `int`. Default value: `0x0`.

SAU_REGION6.NSC

Set NSC for SAU region6 at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION7.BADDR

Base address of SAU region7 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION7.ENABLE

Enable SAU region7 at reset.

Type: `bool`. Default value: `0x0`.

SAU_REGION7.LADDR

Limit address of SAU region7 at reset.

Type: `int`. Default value: `0x0`.

SAU_REGION7.NSC

Set NSC for SAU region7 at reset.

Type: `bool`. Default value: `0x0`.

scheduler_mode

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, 3=ISSCompare) WARNING:

This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `int`. Default value: `0x0`.

SEEXT

Whether the ARMv8-M Security Extensions are included.

Type: `bool`. Default value: `0x1`.

semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x20700000`.

- semihosting-prefix**
Prefix semihosting output with target instance name.
Type: `bool`. Default value: `0x0`.
- semihosting-stack_base**
Virtual address of base of descending stack.
Type: `int`. Default value: `0x20800000`.
- semihosting-stack_limit**
Virtual address of stack limit.
Type: `int`. Default value: `0x20700000`.
- semihosting-Thumb_SVC**
T32 SVC number for semihosting.
Type: `int`. Default value: `0xab`.
- use_architectural_names**
Enable using architectural names for registers LR/PSP/MSP instead of model-specific names.
Type: `bool`. Default value: `0x0`.
- vfp-enable_at_reset**
Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: `bool`. Default value: `0x0`.
- WIC**
Include support for WIC-mode deep sleep.
Type: `bool`. Default value: `0x1`.
- WICLINES**
Number of lines supported by the WIC interface.
Type: `int`. Default value: `0x23`.

4.5.51 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 4-179: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

About ARMCortexM52CT

This model supports the M-Profile Vector Extension (MVE) and the Custom Datapath Extension (CDE). For more information about CDE support, see [5.7 CDE](#) on page 1628.

Ports for ARMCortexM52CT

Table 4-180: Ports

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
ahbp_m	PVBus	Master	The core will generate Vendor System data accesses on this port.
ahbs	PVBus	Slave	Slave AHB - External master (e.g. DMA) can write to TCMs (whether or not enabled in xTCMCR)
auxfault	Value	Slave	This is wired to the Auxiliary Fault Status Register.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions. Referred to in some documents as 'CLKIN'.
coproc_bus	CoprocBusProtocol	Slave	Co-Processor Interface
core_dside_bus_gasket_in	PVBus	Slave	-
core_dside_bus_gasket_out	PVBus	Master	-
cpuwait	Signal	Slave	Stall the CPU out of reset
currns	Signal	Master	Current Security state of the processor
currpri	Value	Master	Current execution priority.
dbgen	Signal	Slave	Invasive debug enable
dbgrestart	Signal	Slave	Request for synchronised exit from halt mode
dbgrestarted	Signal	Master	Handshakes with DBGRESTART
edbgrq	Signal	Slave	External request to enter halt mode
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.
inittcmen[2]	Signal	Slave	Reset configuration port - TCM enable initialisation out of reset Bit[0] HIGH = ITCM is enabled Bit[1] HIGH = DTCM is enabled

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.
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 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
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.

4.5.52 ARM CortexM55CT

CortexM55CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-181: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM CortexM55CT contains the following CADI targets:

- ARM_Cortex-M55

ARM CortexM55CT contains the following MTI components:

- [ARM_Cortex-M55](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)

About ARM CortexM55CT

The model supports Custom Datapath Extension (CDE). For more information, see [5.7 CDE](#) on page 1628.

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_CCNT` is an alias of the `DWT_CYCCNT` register, so they contain the same values. In the model, `PMU_CCNT` 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:

Table 4-182: ITM_PACKET_TYPE field values that the model supports

Field value	Description	Supported by model
<code>ITM_SYNC</code>	Synchronization packet	Not supported.
<code>ITM_P_OVERFLOW</code>	Protocol: Overflow packet	Not supported.
<code>ITM_P_LOCAL_TIMESTAMP</code>	Protocol: Local timestamp packets	Not supported.
<code>ITM_P_GLOBAL_TIMESTAMP</code>	Protocol: Global timestamp packets	Not supported.
<code>ITM_P_EXTEN</code>	Protocol: Extension packet	Not supported.
<code>ITM_S_INSTRUMENTATION</code>	Source: Instrumentation packet	Supported.
<code>ITM_S_DWT_EVENT_COUNTER</code>	Hardware source: Event counter wrapping	Not supported.
<code>ITM_S_DWT_EXCEPTION</code>	Hardware source: Exception tracing	Supported.
<code>ITM_S_DWT_PC_SAMPLING</code>	Hardware source: PC sampling	Not supported.
<code>ITM_S_DWT_DATA_PC_TRACE</code>	Hardware source: DWT Data trace PC value	Supported.
<code>ITM_S_DWT_DATA_ADDRESS_TRACE</code>	Hardware source: DWT Data trace address value	Supported.
<code>ITM_S_DWT_DATA_DATA_TRACE</code>	Hardware source: DWT Data trace DATA value	Supported.

Ports for ARMCortexM55CT

Table 4-183: Ports

Name	Protocol	Type	Description
<code>ahbd</code>	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
<code>ahbp_m</code>	PVBus	Master	The core will generate Vendor System data accesses on this port.
<code>ahbs</code>	PVBus	Slave	Slave AHB - External master (e.g. DMA) can write to TCMs (whether or not enabled in xTCMCR)
<code>auxfault</code>	Value	Slave	This is wired to the Auxiliary Fault Status Register.
<code>bigend</code>	Signal	Slave	Configure big endian data format.
<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. Referred to in some documents as 'CLKIN'.
<code>coproc_bus</code>	CoprocBusProtocol	Slave	Co-Processor Interface
<code>core_dside_bus_gasket_in</code>	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-M55-specific LOCKNSVTOR, LOCKSVTAIRCR, LOCKSMPU, LOCKNSMPU, LOCKSAU. Disable writes to VTOR_NS
lockpahb	Signal	Slave	P-AHB related ports Disable writes to PAHBCR
locksau	Signal	Slave	Disable writes to the SAU_* registers
locksmpu	Signal	Slave	Disable writes to the Secure MPU_* registers

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_Cortex-M55

aircr_iesb_is_writable

IS the AIRCR.IESB bit [5] writable?.

Type: `bool`. Default value: `0x1`.

aircr_iesb_reset

Set the AIRCR.IESB bit [5] after reset.

Type: `bool`. Default value: `0x0`.

BEATS_PER_TICK

Number of beats from each in-flight vector instruction executed in 1 tick (1,2 or 4).

Type: `int`. Default value: `0x2`.

BF_is_nop

BF instruction executes as NOP, even if we have LO_BRANCH_INFO.

Type: `bool`. Default value: `0x0`.

cde_impl_name

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

Type: `string`. Default value: `""`.

CDEMAPPEDONCP

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

Type: `int`. Default value: `0xff`.

CDERTLID

Value of ID_AFR0.CDERTLID.

Type: `int`. Default value: `0x20`.

CFGBIGEND

Initialize processor to big endian mode.

Type: `bool`. Default value: `0x0`.

CFGDTCMSZ

Size of the data TCM. 0=No DTCM implemented. Otherwise=Size of DTCM= $\text{pow}(2, \text{CFGDTCMSZ} - 1)$ KB. Minimum size is 4KB.

Type: `int`. Default value: `0x9`.

CFGITCMSZ

Size of the instruction TCM. 0=No ITCM implemented. Otherwise=Size of ITCM= $\text{pow}(2, \text{CFGITCMSZ} - 1)$ KB. Minimum size is 4KB.

Type: `int`. Default value: `0x9`.

CFGMEMALIAS

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.

Type: `int`. Default value: `0x0`.

CFGNOCDECP

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

Type: `int`. Default value: `0x0`.

CFGPAHBSZ

Size of the P-AHB peripheral port memory region. 0=P-AHB disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

Type: `int`. Default value: `0x0`.

cpi_div

divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

CPIF

Specifies whether the external coprocessor interface is included.

Type: `bool`. Default value: `0x1`.

CPNSPRESENT

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

Type: `int`. Default value: `0xff`.

CPSPRESENT

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

Type: `int`. Default value: `0xff`.

CTI

CTI (Cross Trigger Interface) included.

Type: `bool`. Default value: `0x0`.

CTI_irq0_pin

CTI interrupt request 0 pin.

Type: `int`. Default value: `0x4`.

CTI_irq1_pin

CTI interrupt request 1 pin.

Type: `int`. Default value: `0x5`.

DBG_LVL

0: Minimal debug; 1: 2 Watchpoints, 4 Breakpoint comparators; 2: 4 Watchpoints, 8 Breakpoint comparators; 3: 8 Watchpoints, 8 Breakpoint comparators.

Type: `int`. Default value: `0x2`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-ways

L1 D-cache ways (sets are implicit from size).

Type: `int`. Default value: `0x4`.

DCACHESZ

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.

Type: `int`. Default value: `0xf`.

delay_faultmask_update

Delay FAULTMASK update to context sync.

Type: `bool`. Default value: `0x0`.

delay_sysreg_update

Delay some system register updates (e.g. SHCSR) to context sync.

Type: `bool`. Default value: `0x0`.

DTGU

DTCM Security Gate Unit included.

Type: `bool`. Default value: `0x0`.

DTGUBLKSZ

DTCM gate unit block size. Size= $\text{pow}(2, \text{DTGUBLKSZ} + 5)$ bytes.

Type: `int`. Default value: `0x3`.

DTGUMAXBLKS

Maximum number of DTCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{DTGUMAXBLKS})$.

Type: `int`. Default value: `0x0`.

duplicate_CADI_TCM_writes

CADI writes to TCMs are also sent to downstream memory at same addresses (for validation platforms).

Type: `bool`. Default value: `0x0`.

ecc_on

Enable Error Correcting Code.

Type: `bool`. Default value: `0x0`.

ECOREVNUM

ECO Revision number.

Type: `int`. Default value: `0x0`.

ERRDEVID.NUM

RAS: Number of implemented error record indexes, 0 to 1.

Type: `int`. Default value: `0x1`.

ETM

Support for ETM trace. `false` : No ETM trace included, `true`: ETM trace included.

Type: `bool`. Default value: `0x1`.

execute_via_archex

DEPRECATED: Removed ArchEx-generated code.

Type: `bool`. Default value: `0x1`.

FPU

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

has_cde

Enables Custom Datapath Extensions.

Type: `bool`. Default value: `0x0`.

has_core_dside_bus_gasket

STL gasket enabled.

Type: `bool`. Default value: `0x0`.

has_unprivileged_debug

Unprivileged Debug Extension supported for Mainline Extension.

Type: `bool`. Default value: `0x1`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

icache-ways

L1 I-cache ways (sets are implicit from size).

Type: `int`. Default value: `0x2`.

ICACHESZ

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.

Type: `int`. Default value: `0xF`.

ID_ISAR0.CmpBranch

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.

Type: `int`. Default value: `0x3`.

INITNSVTOR

Non-Secure vector-table offset at reset.

Type: `int`. Default value: `0x0`.

INITPAHBEN

The P-AHB enable state at reset.

Type: `bool`. Default value: `0x0`.

INITSVTOR

Secure vector-table offset at reset.

Type: `int`. Default value: `0x0`.

INITTCMEN

The TCM enable state at reset. Bit 0 corresponds to ITCM enable state, bit 1 corresponds to DTCM enable state.

Type: `int`. Default value: `0x3`.

IRQDIS0

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[n+0].

Type: `int`. Default value: `0x0`.

IRQDIS1

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[n+32].

Type: `int`. Default value: `0x0`.

IRQDIS10

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[n+320].

Type: `int`. Default value: `0x0`.

IRQDIS11

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[n+352].

Type: `int`. Default value: `0x0`.

IRQDIS12

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[n+384].

Type: `int`. Default value: `0x0`.

IRQDIS13

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[n+416].

Type: `int`. Default value: `0x0`.

IRQDIS14

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].
Type: `int`. Default value: `0x0`.

IRQDIS2

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].
Type: `int`. Default value: `0x0`.

IRQDIS3

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].
Type: `int`. Default value: `0x0`.

IRQDIS4

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].
Type: `int`. Default value: `0x0`.

IRQDIS5

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].
Type: `int`. Default value: `0x0`.

IRQDIS6

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].
Type: `int`. Default value: `0x0`.

IRQDIS7

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].
Type: `int`. Default value: `0x0`.

IRQDIS8

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].
Type: `int`. Default value: `0x0`.

IRQDIS9

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].
Type: `int`. Default value: `0x0`.

IRQLVL

Number of bits of interrupt priority.
Type: `int`. Default value: `0x3`.

ITGU

ITCM Security Gate Unit included.
Type: `bool`. Default value: `0x0`.

ITGUBLKSZ

ITCM gate unit block size. Size= $\text{pow}(2, \text{ITGUBLKSZ} + 5)$ bytes.
Type: `int`. Default value: `0x3`.

ITGUMAXBLKS

Maximum number of ITCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{ITGUMAXBLKS})$.
Type: `int`. Default value: `0x0`.

ITM

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

Type: bool. Default value: 0x1.

IWIC

Include support for Internal Wake-up Interrupt Controller.

Type: bool. Default value: 0x1.

LOCK_NS_MPU

Lock down of Non-Secure MPU registers write.

Type: bool. Default value: 0x0.

LOCK_S_MPU

Lock down of Secure MPU registers write.

Type: bool. Default value: 0x0.

LOCK_SAU

Lock down of SAU registers write.

Type: bool. Default value: 0x0.

LOCKDTGU

Lock down of Data TGU registers write.

Type: bool. Default value: 0x0.

LOCKITGU

Lock down of Instruction TGU registers write.

Type: bool. Default value: 0x0.

LOCKTCM

Lock down of TCM registers write.

Type: bool. Default value: 0x0.

master_id

Master ID presented in bus transactions.

Type: int. Default value: 0x0.

min_sync_level

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

Type: int. Default value: 0x0.

MPU_NS

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

Type: int. Default value: 0x8.

MPU_S

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

Type: int. Default value: 0x8.

MVE

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.

Type: int. Default value: 0x1.

mve_unpred_config_json

A JSON array of arrays of the form [unpred_result, instr, cond] for unpredictable configuration for MVE. Either instr or cond may be omitted. The first match wins. Use 'list' to show the available options.

Type: string. Default value: `""[["OK", "VMINNMV_f_T2", "Rda==11x1"], ["OK", "VDDUP", "curOffset MOD imm32 != 0"], ["OK", "VDDUP", "bufSize MOD imm32 != 0"], ["OK", "VDDUP", "curOffset >= bufSize "]]"`.

num_pmu_counters

Number of available PMU counters.

Type: int. Default value: 0x1f.

NUMIRQ

Number of user interrupts.

Type: int. Default value: 0x20.

ras_cei_pin

RAS: Critical error interrupt pin.

Type: int. Default value: 0x2.

ras_cei_support

RAS: Whether Critical Error Interrupt is supported.

Type: bool. Default value: 0x1.

ras_eri_pin

RAS: Error recovery interrupt pin.

Type: int. Default value: 0x1.

ras_eri_support

RAS: Whether Error Recovery Interrupt is supported.

Type: bool. Default value: 0x1.

ras_ERRFR0

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. `{ "ED":0x1, "UE":0x1 }`.

Type: string. Default value: `""{ "ED":0x1, "UE":0x1 }"`.

ras_error_record

1 bit value that specifies which nodes out of 0-1 are implemented (ERRDEVID is derived from this parameter).

Type: int. Default value: 0x1.

ras_fhi_pin

RAS: Fault handling interrupt pin.

Type: int. Default value: 0x0.

ras_fhi_support

RAS: Whether Fault Handling Interrupt is supported.

Type: `bool`. Default value: `0x1`.

SAU

Number of SAU regions (0 => no SAU).

Type: `int`. Default value: `0x4`.

scheduler_mode

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, 3=ISSCompare) WARNING: This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `int`. Default value: `0x0`.

SECEXT

Whether the ARMv8-M Security Extensions are included.

Type: `bool`. Default value: `0x1`.

semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x20700000`.

semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x20800000`.

semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0x20700000`.

- semihosting-Thumb_SVC**
T32 SVC number for semihosting.
Type: `int`. Default value: `0xab`.
- tcm_cadi_accesses_are_physical**
CADI accesses to TCMs ignore any alias regions configured (for validation platforms).
Type: `bool`. Default value: `0x0`.
- tcm_fill_pattern_1**
TCM Fill pattern 1.
Type: `int`. Default value: `0xdfdfdfcf`.
- tcm_fill_pattern_2**
TCM Fill pattern 2.
Type: `int`. Default value: `0xcfdfdfdf`.
- trace_style**
MVE instruction trace style: 0=Tarmac-like from instDB.json, 1=execute function+params (for debug), 2 = Rosetta. Add 16 for `[**--]` beat trace. Add 32 for tracing IMPLIED LOB instructions. Add 64 to change opcode of implied BF to `0xBF00`.
Type: `int`. Default value: `0x0`.
- unpred_config_json_file**
Path to the unpredictable configuration file in JSON format.
Type: `string`. Default value: `""`.
- use_architectural_names**
Enable using architectural names for registers LR/PSP/MSP instead of model-specific names.
Type: `bool`. Default value: `0x0`.
- vfp-enable_at_reset**
Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: `bool`. Default value: `0x0`.
- WICLINES**
Number of lines supported by the WIC interface.
Type: `int`. Default value: `0x23`.

4.5.53 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 4-184: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexM85CT contains the following CADI targets:

- [ARM_Cortex-M85](#)

ARMCortexM85CT contains the following MTI components:

- [ARM_Cortex-M85](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)

About ARMCortexM85CT

This model supports the M-Profile Vector Extension (MVE) and the Custom Datapath Extension (CDE). For more information about CDE support, see [5.7 CDE](#) on page 1628.

Ports for ARMCortexM85CT

Table 4-185: Ports

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
ahbp_m	PVBus	Master	The core will generate Vendor System data accesses on this port.
ahbs	PVBus	Slave	Slave AHB - External master (e.g. DMA) can write to TCMs (whether or not enabled in xTCMCR)
auxfault	Value	Slave	This is wired to the Auxiliary Fault Status Register.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions. Referred to in some documents as 'CLKIN'.
coproc_bus	CoprocBusProtocol	Slave	Co-Processor Interface
core_dside_bus_gasket_in	PVBus	Slave	-
core_dside_bus_gasket_out	PVBus	Master	-
cpuwait	Signal	Slave	Stall the CPU out of reset
currns	Signal	Master	Current Security state of the processor
currpri	Value	Master	Current execution priority.
dbgen	Signal	Slave	Invasive debug enable
dbgrestart	Signal	Slave	Request for synchronised exit from halt mode
dbgrestarted	Signal	Master	Handshakes with DBGRESTART
edbgrq	Signal	Slave	External request to enter halt mode
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.

Name	Protocol	Type	Description
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.
inittcmen[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 PAHBSCR
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
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	-

Name	Protocol	Type	Description
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
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_Cortex-M85

aircr_iesb_is_writable

IS the AIRCR.IESB bit [5] writable?.

Type: `bool`. Default value: `0x1`.

aircr_iesb_reset

Set the AIRCR.IESB bit [5] after reset.

Type: `bool`. Default value: `0x0`.

BEATS_PER_TICK

Number of beats from each in-flight vector instruction executed in 1 tick (1,2 or 4).

Type: `int`. Default value: `0x2`.

BF_is_nop

BF instruction executes as NOP, even if we have LO_BRANCH_INFO.

Type: `bool`. Default value: `0x0`.

cde_impl_name

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

Type: `string`. Default value: `""`.

CDEMAPPEDONCP

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

Type: `int`. Default value: `0xff`.

CDERTLID

Value of ID_AFR0.CDERTLID.

Type: `int`. Default value: `0x20`.

CFGBIGEND

Initialize processor to big endian mode.

Type: `bool`. Default value: `0x0`.

CFGCPUINST

CPU instance number. This is part of the TCM base address, in bits 25:24.

Type: `int`. Default value: `0x0`.

CFGDTCMSZ

Size of the data TCM. 0=No DTCM implemented. Otherwise=Size of DTCM=pow(2, CFGDTCMSZ - 1) KB. Minimum size is 4KB.

Type: `int`. Default value: `0x9`.

CFGITCMSZ

Size of the instruction TCM. 0=No ITCM implemented. Otherwise=Size of ITCM=pow(2, CFGITCMSZ - 1) KB. Minimum size is 4KB.

Type: `int`. Default value: `0x9`.

CFGMEMALIAS

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.

Type: `int`. Default value: `0x0`.

CFGNOCDECP

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

Type: `int`. Default value: `0x0`.

CFGPACBTI

Enables support for the Pointer Authentication and Branch Target Identification (PACBTI)

Extension. FALSE: Disabled, TRUE:PAC implemented using the QARMA5 algorithm with BTI.

Type: `bool`. Default value: `0x0`.

CFGPAHBSZ

Size of the P-AHB peripheral port memory region. 0=P-AHB disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

Type: `int`. Default value: `0x0`.

cpi_div

divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

CPIF

Specifies whether the external coprocessor interface is included.

Type: `bool`. Default value: `0x1`.

CPNSPRESENT

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

Type: `int`. Default value: `0xff`.

CPSPRESENT

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

Type: `int`. Default value: `0xff`.

CTI

CTI (Cross Trigger Interface) included.

Type: `bool`. Default value: `0x0`.

CTI_irq0_pin

CTI interrupt request 0 pin.

Type: `int`. Default value: `0x4`.

CTI_irq1_pin

CTI interrupt request 1 pin.

Type: `int`. Default value: `0x5`.

DBGVLV

1: 4 Watchpoints, 4 Breakpoint comparators; 2: 8 Watchpoints, 8 Breakpoint comparators.

Type: `int`. Default value: `0x2`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-ways

L1 D-cache ways (sets are implicit from size).

Type: `int`. Default value: `0x4`.

DCACHESZ

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.

Type: `int`. Default value: `0xf`.

delay_faultmask_update

Delay FAULTMASK update to context sync.

Type: `bool`. Default value: `0x0`.

delay_sysreg_update

Delay some system register updates (e.g. SHCSR) to context sync.

Type: `bool`. Default value: `0x0`.

DTGU

DTCM Security Gate Unit included.

Type: `bool`. Default value: `0x0`.

DTGUBLKSZ

DTCM gate unit block size. Size= $\text{pow}(2, \text{DTGUBLKSZ} + 5)$ bytes.

Type: `int`. Default value: `0x3`.

DTGUMAXBLKS

Maximum number of DTCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{DTGUMAXBLKS})$.

Type: `int`. Default value: `0x0`.

duplicate_CADI_TCM_writes

CADI writes to TCMs are also sent to downstream memory at same addresses (for validation platforms).

Type: `bool`. Default value: `0x0`.

ecc_on

Enable Error Correcting Code.

Type: `bool`. Default value: `0x0`.

ECOREVNUM

ECO Revision number.

Type: `int`. Default value: `0x0`.

ERRDEVID.NUM

RAS: Number of implemented error record indexes, 0 to 1.

Type: `int`. Default value: `0x1`.

ETM

Support for ETM trace. `false` : No ETM trace included, `true`: ETM trace included.

Type: `bool`. Default value: `0x1`.

execute_via_archex

DEPRECATED: Removed ArchEx-generated code.

Type: `bool`. Default value: `0x1`.

FPU

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

has_cde

Enables Custom Datapath Extensions.

Type: `bool`. Default value: `0x0`.

has_core_dside_bus_gasket

STL gasket enabled.

Type: `bool`. Default value: `0x0`.

has_unprivileged_debug

Unprivileged Debug Extension supported for Mainline Extension.

Type: `bool`. Default value: `0x1`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

icache-ways

L1 I-cache ways (sets are implicit from size).

Type: `int`. Default value: `0x2`.

ICACHESZ

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.

Type: `int`. Default value: `0xf`.

ID_ISAR0.CmpBranch

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.

Type: `int`. Default value: `0x3`.

ID_ISAR5.PACBTI

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.

Type: `int`. Default value: `0x0`.

INITNSVTOR

Non-Secure vector-table offset at reset.

Type: `int`. Default value: `0x0`.

INITPAHBEN

The P-AHB enable state at reset.

Type: `bool`. Default value: `0x0`.

INITSVTOR

Secure vector-table offset at reset.

Type: `int`. Default value: `0x0`.

INITTCMEN

The TCM enable state at reset. Bit 0 corresponds to ITCM enable state, bit 1 corresponds to DTCM enable state.

Type: `int`. Default value: `0x3`.

IRQDIS0

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+0].

Type: `int`. Default value: `0x0`.

IRQDIS1

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+32].

Type: `int`. Default value: `0x0`.

IRQDIS10

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+320].

Type: `int`. Default value: `0x0`.

IRQDIS11

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+352].

Type: `int`. Default value: `0x0`.

IRQDIS12

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+384].

Type: `int`. Default value: `0x0`.

IRQDIS13

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+416].

Type: `int`. Default value: `0x0`.

IRQDIS14

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+448]`.
Type: `int`. Default value: `0x0`.

IRQDIS2

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+64]`.
Type: `int`. Default value: `0x0`.

IRQDIS3

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+96]`.
Type: `int`. Default value: `0x0`.

IRQDIS4

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+128]`.
Type: `int`. Default value: `0x0`.

IRQDIS5

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+160]`.
Type: `int`. Default value: `0x0`.

IRQDIS6

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+192]`.
Type: `int`. Default value: `0x0`.

IRQDIS7

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+224]`.
Type: `int`. Default value: `0x0`.

IRQDIS8

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+256]`.
Type: `int`. Default value: `0x0`.

IRQDIS9

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+288]`.
Type: `int`. Default value: `0x0`.

IRQLVL

Number of bits of interrupt priority.
Type: `int`. Default value: `0x3`.

iss_env

Enable ISSComp logic.
Type: `bool`. Default value: `0x0`.

ITGU

ITCM Security Gate Unit included.
Type: `bool`. Default value: `0x0`.

ITGUBLKSZ

ITCM gate unit block size. $\text{Size} = \text{pow}(2, \text{ITGUBLKSZ} + 5)$ bytes.
Type: `int`. Default value: `0x3`.

ITGUMAXBLKS

Maximum number of ITCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{ITGUMAXBLKS})$.

Type: `int`. Default value: `0x0`.

ITM

Level of instrumentation trace supported. `false` : No ITM trace included, `true`: ITM trace included.

Type: `bool`. Default value: `0x1`.

IWIC

Include support for Internal Wake-up Interrupt Controller.

Type: `bool`. Default value: `0x1`.

LOCK_NS_MPU

Lock down of Non-Secure MPU registers write.

Type: `bool`. Default value: `0x0`.

LOCK_S_MPU

Lock down of Secure MPU registers write.

Type: `bool`. Default value: `0x0`.

LOCK_SAU

Lock down of SAU registers write.

Type: `bool`. Default value: `0x0`.

LOCKDTGU

Lock down of Data TGU registers write.

Type: `bool`. Default value: `0x0`.

LOCKITGU

Lock down of Instruction TGU registers write.

Type: `bool`. Default value: `0x0`.

LOCKTCM

Lock down of TCM registers write.

Type: `bool`. Default value: `0x0`.

master_id

Master ID presented in bus transactions.

Type: `int`. Default value: `0x0`.

min_sync_level

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

Type: `int`. Default value: `0x0`.

MPU_NS

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

Type: `int`. Default value: `0x8`.

MPU_S

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

Type: `int`. Default value: `0x8`.

MVE

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.

Type: int. Default value: 0x1.

mve_unpred_config_json

A JSON array of arrays of the form [unpred_result, instr, cond] for unpredictable configuration for MVE. Either instr or cond may be omitted. The first match wins. Use 'list' to show the available options.

Type: string. Default value: `""[["OK", "VMINNMV_f_T2", "Rda==11x1"], ["OK", "VDDUP", "curOffset MOD imm32 != 0"], ["OK", "VDDUP", "bufSize MOD imm32 != 0"], ["OK", "VDDUP", "curOffset >= bufSize "]]"`.

num_pmu_counters

Number of available PMU counters.

Type: int. Default value: 0x1f.

NUMIRQ

Number of user interrupts.

Type: int. Default value: 0x20.

ras_cei_pin

RAS: Critical error interrupt pin.

Type: int. Default value: 0x2.

ras_cei_support

RAS: Whether Critical Error Interrupt is supported.

Type: bool. Default value: 0x1.

ras_eri_pin

RAS: Error recovery interrupt pin.

Type: int. Default value: 0x1.

ras_eri_support

RAS: Whether Error Recovery Interrupt is supported.

Type: bool. Default value: 0x1.

ras_ERRFR0

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. `{"ED":0x1,"UE":0x1}`.

Type: string. Default value: `""{"ED":0x1,"UE":0x1}"`.

ras_error_record

1 bit value that specifies which nodes out of 0-1 are implemented (ERRDEVID is derived from this parameter).

Type: int. Default value: 0x1.

ras_fhi_pin

RAS: Fault handling interrupt pin.

Type: int. Default value: 0x0.

ras_fhi_support

RAS: Whether Fault Handling Interrupt is supported.

Type: `bool`. Default value: `0x1`.

SAU

Number of SAU regions (0 => no SAU).

Type: `int`. Default value: `0x4`.

scheduler_mode

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, 3=ISSCompare) WARNING: This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `int`. Default value: `0x0`.

SECEXT

Whether the ARMv8-M Security Extensions are included.

Type: `bool`. Default value: `0x1`.

semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x20700000`.

semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x20800000`.

semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0x20700000`.

semihosting-Thumb_SVC

T32 SVC number for semihosting.

Type: `int`. Default value: `0xab`.**tcm_cadi_accesses_are_physical**

CADI accesses to TCMs ignore any alias regions configured (for validation platforms).

Type: `bool`. Default value: `0x0`.**tcm_fill_pattern_1**

TCM Fill pattern 1.

Type: `int`. Default value: `0xdfdfdfcf`.**tcm_fill_pattern_2**

TCM Fill pattern 2.

Type: `int`. Default value: `0xcfdfdfdf`.**trace_style**MVE instruction trace style: 0=Tarmac-like from instDB.json, 1=execute function+params (for debug), 2 = Rosetta. Add 16 for `[**--]` beat trace. Add 32 for tracing IMPLIED LOB instructions. Add 64 to change opcode of implied BF to `0xBF00`.Type: `int`. Default value: `0x0`.**unpred_config_json_file**

Path to the unpredictable configuration file in JSON format.

Type: `string`. Default value: `""`.**use_architectural_names**

Enable using architectural names for registers LR/PSP/MSP instead of model-specific names.

Type: `bool`. Default value: `0x0`.**vfp-enable_at_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.**WICLINES**

Number of lines supported by the WIC interface.

Type: `int`. Default value: `0x23`.

4.5.54 ARMCortexR4CT

ARMCortexR4CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-186: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexR4CT contains the following CADI targets:

- ARM_Cortex-R4
- PVCache

ARMCortexR4CT contains the following MTI components:

- [ARM_Cortex-R4](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)

About ARMCortexR4CT

- The model implements the `cfgie` port, although it is optional in hardware.
- `pvbus_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 4-187: 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.
<code>ticks</code>	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.
<code>vic_ack</code>	Signal	Master	Vic acknowledge port to primary VIC.
<code>vic_addr</code>	Value	Slave	Vic address port from primary VIC.

Name	Protocol	Type	Description
vinithi	Signal	Slave	Configure high vectors after a reset.

Parameters for ARM_Cortex-R4

CFGEND0

Initialize to BE8 endianness.

Type: `bool`. Default value: `0x0`.

CFGIE

Set the reset value of the instruction endian bit.

Type: `bool`. Default value: `0x0`.

CFGNMFI

Enable nonmaskable FIQ interrupts on startup.

Type: `bool`. Default value: `0x0`.

CFGTE

Initialize to take exceptions in T32 state. Model starts in T32 state.

Type: `bool`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-size

Set D-cache size in bytes.

Type: `int`. Default value: `0x10000`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dtcm0_base

Base address of DTCM at startup.

Type: `int`. Default value: `0x800000`.

dtcm0_size

Size of DTCM in KB.

Type: `int`. Default value: `0x8`.

icache-size

Set I-cache size in bytes.

Type: `int`. Default value: `0x10000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

implements_vfp

Set whether the model has been built with VFP support.

Type: `bool`. Default value: `0x1`.

INITRAMD

Set or reset the INITRAMD signal.

Type: `bool`. Default value: `0x0`.

INITRAMI

Set or reset the INITRAMI signal.

Type: `bool`. Default value: `0x0`.

itcm0_base

Base address of ITCM at startup.

Type: `int`. Default value: `0x0`.

itcm0_size

Size of ITCM in KB.

Type: `int`. Default value: `0x8`.

LOCZRAMI

Set or reset the LOCZRAMI signal.

Type: `bool`. Default value: `0x0`.

master_id

Master ID presented in bus transactions.

Type: `int`. Default value: `0x0`.

min_sync_level

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

Type: `int`. Default value: `0x0`.

NUM_MPU_REGION

Number of MPU regions.

Type: `int`. Default value: `0x8`.

scheduler_mode

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!.

Type: `int`. Default value: `0x0`.

semihosting-ARM_HLT

ARM HLT number for semihosting.

Type: `int`. Default value: `0x1000`.

semihosting-ARM_SVC

ARM SVC number for semihosting.

Type: `int`. Default value: `0x123456`.

semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x1000000`.

semihosting-hlt-enable

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: `bool`. Default value: `0x0`.

semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0x10000000`.

semihosting-Thumb_HLT

Thumb HLT number for semihosting.

Type: `int`. Default value: `0x3c`.

semihosting-Thumb_SVC

Thumb SVC number for semihosting.

Type: `int`. Default value: `0xab`.

vfp-enable_at_reset

Enable coprocessor access and VFP at reset.

Type: `bool`. Default value: `0x0`.

VINITHI

Initialize with high vectors enabled.

Type: `bool`. Default value: `0x0`.

4.5.55 ARM CortexR5x1CT

ARM CortexR5x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-188: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM CortexR5x1CT contains the following CADI targets:

- ARM_Cortex-R5
- Cluster_ARM_Cortex-R5

ARM CortexR5x1CT contains the following MTI components:

- [ARM_Cortex-R5](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)

About ARM CortexR5x1CT

- An ARM CortexR5x2CT component also exists.
- The per-core parameters are preceded by `cpu n` , 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.
- `pvbbus_s` is the slave port to access the TCM RAM of CPU n . Bits [3:0] of the user flags in the transaction are used to select the TCM:
 - 1 selects the ATCM of CPU 0.
 - 2 selects the BTCM of CPU 0.
 - 3 selects the ATCM of CPU 1.

- 4 selects the BTCM of CPU 1.
- Any other value is reserved.

Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- The RR bit in the SCTLR is ignored.
- The Low Latency Peripheral Port is not modeled.
- The model only has a single bus master port combining instruction, data, DMA and peripheral accesses. The CP15 control registers associated with peripheral buses preserve values but do not have any other effect.
- The model only supports static split lock and not dynamic split lock. Contact Arm for details.
- TCMs are modeled internally and the model does not support external TCMs or the ports associated with them.
- The model cannot experience an ECC error and does not support fault injection into the system, so Arm does not provide the ability to set error schemes for the caches or TCMs. Contact Arm if you require a particular value in the Build Options registers.

Vectored Interrupt Controller (VIC) ports

The ARMCortexR4CT and ARMCortexR5xnCT models implement a simplified model of the Vectored Interrupt Controller (VIC) port.

The protocol consists of two ports:

- The `vic_addr` port signals the vectored interrupt address from the external VIC.
- The `vic_ack` port signals the VIC that an interrupt has been detected and is being serviced.

The expected interrupt sequence is:

1. The software enables the VIC interface by setting the VE bit in the CP15 control register and setting up suitable interrupt routines.
2. The VIC asserts IRQ.
3. Some time later, the processor detects and responds to the IRQ by asserting `vic_ack`.
4. The VIC writes the vector address to the processor using `vic_addr`.
5. The processor de-asserts `vic_ack`.
6. The processor transfers control to the vectored address returned from the VIC.

The interaction between the processor and the VIC is untimed after the processor acknowledges the interrupt, so certain interrupt sequences cannot occur in the code translation processor models.

Ports for ARMCortexR5x1CT

Table 4-189: Ports

Name	Protocol	Type	Description
acp_s	PVBus	Slave	ACP slave port.
cfgatcmsz[2]	Value	Slave	ATCM size.
cfgbtcmsz[2]	Value	Slave	BTCM Size.
cfgend[2]	Signal	Slave	This signal is for EE bit initialisation. This is CFGEE in RTL but cfgend here fastsim consistency reasons.
cfgnmfi[2]	Signal	Slave	Controls non-maskable Fast Interrupts.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cpuhalt[2]	Signal	Slave	Raising this signal will put the core into halt mode.
event[2]	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[2]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
groupid	Value	Slave	Group ID used for MPIDR.
initrama[2]	Signal	Slave	If ATCM is enabled at reset.
initramb[2]	Signal	Slave	If BTCM is enabled at reset.
irq[2]	Signal	Slave	This signal drives the CPU's interrupt handling.
loczrama[2]	Signal	Slave	Location of ATCM at reset.
pmuirq[2]	Signal	Master	Interrupt signal from performance monitoring unit.
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
pvbus_s[1]	PVBus	Slave	tcm slave port.
reset[2]	Signal	Slave	Raising this signal will put the core into reset mode.
standbywfe[2]	Signal	Master	This signal indicate if a core is in wfe state RTL calls this WFEPIPESTOPPED.
standbywfi[2]	Signal	Master	This signal indicates if a core is in WFI state RTL uses WFIPIPESTOPPED.
teinit[2]	Signal	Slave	Default exception handling state.
ticks[2]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vic_ack[2]	Signal	Master	Vic acknowledge port to primary VIC.
vic_addr[2]	Value	Slave	Vic address port from primary VIC.
vinithi[2]	Signal	Slave	This signal controls of the location of the exception vectors at reset.

Parameters for ARM_Cortex-R5

cpu0.atcm_base

Model-specific. Sets the base address of the ATCM (forced to 0 if LOCZRAMA is 1).

Type: int. Default value: 0x40000000.

cpu0.btcn_base

Model-specific. Sets the base address of the BTCM (forced to 0 if LOCZRAMA is 0).

Type: int. Default value: 0x0.

cpu0.CFGATCMSZ

Sets the size of the ATCM.

Type: `int`. Default value: `0xe`.

`cpu0.CFGBTCMSZ`

Sets the size of the BTCM.

Type: `int`. Default value: `0xe`.

`cpu0.CFGEND`

Initialize to BE8 endianness.

Type: `bool`. Default value: `0x0`.

`cpu0.CFGIE`

Set the reset value of the instruction endian bit.

Type: `bool`. Default value: `0x0`.

`cpu0.CFGNMFI`

Enable nonmaskable FIQ interrupts on startup.

Type: `bool`. Default value: `0x0`.

`cpu0.dcache-size`

Set D-cache size in bytes.

Type: `int`. Default value: `0x10000`.

`cpu0.DP_FLOAT`

Sets whether double-precision instructions are available.

Type: `bool`. Default value: `0x1`.

`cpu0.icache-size`

Set I-cache size in bytes.

Type: `int`. Default value: `0x10000`.

`cpu0.INITRAMA`

Initialize with TCMA enabled.

Type: `bool`. Default value: `0x0`.

`cpu0.INITRAMB`

Initialize with TCMB enabled.

Type: `bool`. Default value: `0x0`.

`cpu0.LOCZRAMA`

Initialize with LOCZRAMA set to 1.

Type: `bool`. Default value: `0x0`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-ARM_HLT`

ARM HLT number for semihosting.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

ARM SVC number for semihosting.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-hlt-enable

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: `bool`. Default value: `0x0`.

cpu0.semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-Thumb_HLT

Thumb HLT number for semihosting.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

Thumb SVC number for semihosting.

Type: `int`. Default value: `0xab`.

cpu0.TEINIT

T32 exception enable. The default has exceptions including reset handled in A32 state.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-enable_at_reset

Enable coprocessor access and VFP at reset.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether model has VFP support.
Type: `bool`. Default value: `0x1`.

cpu0.VINITHI

Initialize with high vectors enabled.
Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-R5**cpi_div**

Divider for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).
Type: `int`. Default value: `0x1`.

dcache-state_modelled

Set whether D-cache has stateful implementation.
Type: `bool`. Default value: `0x0`.

GROUP_ID

Value read in GROUP ID register field, bits[15:8] of the MPIDR.
Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.
Type: `bool`. Default value: `0x0`.

INST_ENDIAN

Controls whether the model supports the instruction endianness bit.
Type: `bool`. Default value: `0x1`.

LOCK_STEP

Affects dual-processor configurations only, and ignored by single-processor configurations.
Type: `int`. Default value: `0x0`.

MICRO_SCU

Controls whether the effects of the MicroSCU are modeled.
Type: `bool`. Default value: `0x1`.

NUM_BREAKPOINTS

Controls with how many breakpoint pairs the model has been configured. This only affects the build options registers, because debug is not modeled.
Type: `int`. Default value: `0x3`.

NUM_MPU_REGION

Sets the number of MPU regions.
Type: `int`. Default value: `0xc`.

NUM_WATCHPOINTS

Controls with how many watchpoint pairs the model has been configured. This only affects the build options registers, because debug is not modeled.

Type: `int`. Default value: `0x2`.

scheduler_mode

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!.

Type: `int`. Default value: `0x0`.

SLSPLIT

Sets whether the model starts in split mode or locked mode.

Type: `bool`. Default value: `0x0`.

4.5.56 ARM CortexR7x1CT

ARM CortexR7x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-190: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM CortexR7x1CT contains the following CADI targets:

- `ARM_Cortex-R7`
- `Cluster_ARM_Cortex-R7`
- `PVCache`

ARM CortexR7x1CT contains the following MTI components:

- [ARM_Cortex-R7](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)

About ARM CortexR7x1CT

- An ARM CortexR7x2CT component also exists.

- The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0 or 1).
- `pvbuss` 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 4-191: Ports

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
cfgend[1]	Signal	Slave	This signal is for EE bit initialisation.
cfgnmfi[1]	Signal	Slave	This signal disables FIQ mask in CPSR.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[1]	Signal	Slave	Legacy FIQ request input line.
fiqout[1]	Signal	Master	Output of individual processor nFIQ from the interrupt controller.
fpuflags[1]	ValueState	Master	Floating-Point Unit output flags.
halt[1]	Signal	Slave	Raising this signal will put the core into halt mode. Equivalent to the hardware nCPUHALT[N:0] signal.
initram[1]	Signal	Slave	This signal enables the processor to boot from the instruction TCM.
ints[480]	Signal	Slave	Interrupt distributor interrupt lines.
irq[1]	Signal	Slave	Legacy IRQ request input line.
irqout[1]	Signal	Master	Output of individual processor nIRQ from the interrupt controller.
mfilteren	Signal	Slave	This signal enables filtering of address ranges between master bus ports.
mfilterend	Value	Slave	This port sets end of region mapped to pvbus_m1.
mfilterstart	Value	Slave	This port sets start of region mapped to pvbus_m1.
periphbase	Value	Slave	This port sets the base address of private peripheral region.
periphclk_in	ClockSignal	Slave	The timer and the watchdog take need a clk that is scaled down at least by factor of two.
periphreset	Signal	Slave	This signal resets timer and interrupt controller.
pfilterend	Value	Slave	This port sets end of region mapped to pvbus_mp.
pfilterstart	Value	Slave	This port sets start of region mapped to pvbus_mp.
pmuirq[2]	Signal	Master	Interrupt signal from performance monitoring unit.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_m1	PVBus	Master	The core will generate bus requests on this port.
pvbus_mp	PVBus	Master	The core will generate bus requests on this port.
pvbus_s	PVBus	Slave	tcm slave port
reset[1]	Signal	Slave	Raising this signal will put the core into reset mode.
scureset	Signal	Slave	This signal resets SCU.
smpnamp[1]	Signal	Master	This signals AMP or SMP mode for each Cortex-R7 processor.
standbywfe[1]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[1]	Signal	Master	This signal indicates if a core is in WFI state.
teinit[1]	Signal	Slave	This signal provides default exception handling state.
ticks[1]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.

Name	Protocol	Type	Description
vinithi[1]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
wdreset[1]	Signal	Slave	This signal resets individual watchdog.
wdresetreq[1]	Signal	Master	CPU watchdog reset requests.

Parameters for ARM_Cortex-R7

cpu0.CFGEND

Initialize to BE8 endianness.

Type: bool. Default value: 0x0.

cpu0.CFGNMFI

Enable nonmaskable FIQ interrupts on startup.

Type: bool. Default value: 0x0.

cpu0.dcache-size

Set D-cache size in bytes.

Type: int. Default value: 0x8000.

cpu0.DP_FLOAT

Sets whether double-precision instructions are available.

Type: bool. Default value: 0x1.

cpu0.dtcn_size

DTCM size in KB.

Type: int. Default value: 0x8.

cpu0.icache-size

Set I-cache size in bytes.

Type: int. Default value: 0x8000.

cpu0.INITRAM

Enable the processor to boot from the instruction TCM.

Type: bool. Default value: 0x0.

cpu0.itcm_size

ITCM size in KB.

Type: int. Default value: 0x8.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.POWERCTLI

Default power control state for processor.

Type: int. Default value: 0x0.

cpu0.semihosting-ARM_HLT

ARM HLT number for semihosting.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

ARM SVC number for semihosting.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-hlt-enable

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: `bool`. Default value: `0x0`.

cpu0.semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-Thumb_HLT

Thumb HLT number for semihosting.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

Thumb SVC number for semihosting.

Type: `int`. Default value: `0xab`.

cpu0.SMPnAMP

Set whether the processor is part of a coherent domain.

Type: `bool`. Default value: `0x0`.

cpu0.tcm-present

Disables the DTCM and ITCM.
Type: bool. Default value: 0x1.

cpu0.TEINIT

T32 exception enable. The default has exceptions including reset handled in A32 state.
Type: bool. Default value: 0x0.

cpu0.vfp-enable_at_reset

Enable coprocessor access and VFP at reset.
Type: bool. Default value: 0x0.

cpu0.vfp-present

Set whether model has VFP support.
Type: bool. Default value: 0x1.

cpu0.VINITHI

Initialize with high vectors enabled.
Type: bool. Default value: 0x0.

Parameters for Cluster_ARM_Cortex-R7**CLUSTER_ID**

Processor cluster ID value.
Type: int. Default value: 0x0.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).
Type: int. Default value: 0x1.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).
Type: int. Default value: 0x1.

dcache-state_modelled

Set whether D-cache has stateful implementation.
Type: bool. Default value: 0x0.

dic-spi_count

Number of shared peripheral interrupts implemented.
Type: int. Default value: 0x40.

ecc_on

Enable Error Correcting Code.
Type: bool. Default value: 0x0.

icache-state_modelled

Set whether I-cache has stateful implementation.
Type: bool. Default value: 0x0.

LOCK_STEP

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.

Type: `int`. Default value: `0x0`.

MFILTEREN

Enables filtering of address ranges.

Type: `bool`. Default value: `0x0`.

MFILTEREND

Specifies the end address for address filtering.

Type: `int`. Default value: `0x0`.

MFILTERSTART

Specifies the start address for address filtering.

Type: `int`. Default value: `0x0`.

NUM_MPU_REGION

Sets the number of MPU regions.

Type: `int`. Default value: `0xc`.

PERIPHBASE

Base address of peripheral memory space.

Type: `int`. Default value: `0xae000000`.

PFILTEREND

Specifies the end address for peripheral port address filtering.

Type: `int`. Default value: `0x0`.

PFILTERSTART

Specifies the start address for peripheral port address filtering.

Type: `int`. Default value: `0xffff0000`.

scheduler_mode

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!.

Type: `int`. Default value: `0x0`.

4.5.57 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 4-192: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexR8x1CT contains the following CADI targets:

- ARM_Cortex-R8
- Cluster_ARM_Cortex-R8
- PVCache

ARMCortexR8x1CT contains the following MTI components:

- [ARM_Cortex-R8](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [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 4-193: Ports

Name	Protocol	Type	Description
<code>acp_s</code>	PVBus	Slave	AXI ACP slave port.
<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.
<code>event</code>	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
<code>fiq[1]</code>	Signal	Slave	Legacy FIQ request input line.
<code>fiqout[1]</code>	Signal	Master	Output of individual processor nFIQ from the interrupt controller.
<code>fpfilterend0</code>	Value	Slave	This port sets end of region mapped to <code>pvbus_mfp0</code> .

Name	Protocol	Type	Description
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.
standbywfi[1]	Signal	Master	This signal indicates if a core is in WFI state.
teinit[1]	Signal	Slave	This signal provides default exception handling state.
ticks[1]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.

Name	Protocol	Type	Description
vinithi[1]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
wdreset[1]	Signal	Slave	This signal resets individual watchdog.
wdresetreq[1]	Signal	Master	CPU watchdog reset requests.

Parameters for ARM_Cortex-R8

cpu0.CFGEND

Initialize to BE8 endianness.

Type: bool. Default value: 0x0.

cpu0.CFGNMFI

Enable nonmaskable FIQ interrupts on startup.

Type: bool. Default value: 0x0.

cpu0.dcache-size

Set D-cache size in bytes.

Type: int. Default value: 0x8000.

cpu0.DP_FLOAT

Sets whether double-precision instructions are available.

Type: bool. Default value: 0x1.

cpu0.dtcn_size

DTCM size in KB.

Type: int. Default value: 0x8.

cpu0.icache-size

Set I-cache size in bytes.

Type: int. Default value: 0x8000.

cpu0.INITRAM

Enable the processor to boot from the instruction TCM.

Type: bool. Default value: 0x0.

cpu0.itcm_size

ITCM size in KB.

Type: int. Default value: 0x8.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.POWERCTLI

Default power control state for processor.

Type: int. Default value: 0x0.

cpu0.semihosting-ARM_HLT

ARM HLT number for semihosting.

Type: int. Default value: 0x1000.

cpu0.semihosting-ARM_SVC

ARM SVC number for semihosting.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-hlt-enable

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: `bool`. Default value: `0x0`.

cpu0.semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-Thumb_HLT

Thumb HLT number for semihosting.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

Thumb SVC number for semihosting.

Type: `int`. Default value: `0xab`.

cpu0.SMPnAMP

Set whether the processor is part of a coherent domain.

Type: `bool`. Default value: `0x0`.

cpu0.tcm-present

Disables the DTCM and ITCM.

Type: bool. Default value: 0x1.

cpu0.TEINIT

T32 exception enable. The default has exceptions including reset handled in A32 state.

Type: bool. Default value: 0x0.

cpu0.vfp-enable_at_reset

Enable coprocessor access and VFP at reset.

Type: bool. Default value: 0x0.

cpu0.vfp-present

Set whether model has VFP support.

Type: bool. Default value: 0x1.

cpu0.VINITHI

Initialize with high vectors enabled.

Type: bool. Default value: 0x0.

Parameters for Cluster_ARM_Cortex-R8**CLUSTER_ID**

Processor cluster ID value.

Type: int. Default value: 0x0.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: int. Default value: 0x1.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: int. Default value: 0x1.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: bool. Default value: 0x0.

dic-spi_count

Number of shared peripheral interrupts implemented.

Type: int. Default value: 0x40.

ecc_on

Enable Error Correcting Code.

Type: bool. Default value: 0x0.

FPFILTEREND0

Specifies the end address for the fast peripheral port address filtering.

Type: int. Default value: 0x0.

FPFILTEREND1

Specifies the end address for the fast peripheral port address filtering.

Type: int. Default value: 0x0.

FPFILTEREND2

Specifies the end address for the fast peripheral port address filtering.

Type: `int`. Default value: `0x0`.

FPFILTEREND3

Specifies the end address for the fast peripheral port address filtering.

Type: `int`. Default value: `0x0`.

FPFILTERSTART0

Specifies the start address for the fast peripheral port address filtering.

Type: `int`. Default value: `0xffff00000`.

FPFILTERSTART1

Specifies the start address for the fast peripheral port address filtering.

Type: `int`. Default value: `0xffff00000`.

FPFILTERSTART2

Specifies the start address for the fast peripheral port address filtering.

Type: `int`. Default value: `0xffff00000`.

FPFILTERSTART3

Specifies the start address for the fast peripheral port address filtering.

Type: `int`. Default value: `0xffff00000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

LOCK_STEP

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.

Type: `int`. Default value: `0x0`.

MFILTEREN

Enables filtering of address ranges.

Type: `bool`. Default value: `0x0`.

MFILTEREND

Specifies the end address for address filtering.

Type: `int`. Default value: `0x0`.

MFILTERSTART

Specifies the start address for address filtering.

Type: `int`. Default value: `0x0`.

NUM_MPU_REGION

Sets the number of MPU regions.

Type: `int`. Default value: `0xc`.

PERIPHBASE

Base address of peripheral memory space.
Type: `int`. Default value: `0xae000000`.

PFILTEREND

Specifies the end address for peripheral port address filtering.
Type: `int`. Default value: `0x0`.

PFILTERSTART

Specifies the start address for peripheral port address filtering.
Type: `int`. Default value: `0xffffffff`.

scheduler_mode

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!.
Type: `int`. Default value: `0x0`.

4.5.58 **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 4-194: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexR52PlusCT contains the following CADI targets:

- `ARM_Cortex-R52Plus`
- `Cluster_ARM_Cortex-R52Plus`
- `PVCache`
- `TlbCadi`
- `gic_iri`

ARMCortexR52PlusCT contains the following MTI components:

- [ARM_Cortex-R52Plus](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [GICv3CPUInterface](#)
- [GICv3CPUInterfaceDecoder](#)

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

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.

The following models also exist:

- [ARMCortex-R52Plusx1CT](#)
- [ARMCortex-R52Plusx2CT](#)
- [ARMCortex-R52Plusx3CT](#)
- [ARMCortex-R52Plusx4CT](#)

Ports for ARMCortexR52PlusCT

Table 4-195: Ports

Name	Protocol	Type	Description
<code>cfgdbgromaddr</code>	Value_64	Slave	Debug ROM base address.
<code>cfgdbgromaddrv</code>	Signal	Slave	Debug ROM base address valid.
<code>cfgendianess[4]</code>	Signal	Slave	This signal if for EE bit initialisation.
<code>cfgperiphbase</code>	Value_64	Slave	This port sets the base address of private peripheral region.
<code>cfgthumbexceptions[4]</code>	Signal	Slave	This signal provides default exception handling state.
<code>cfgvectable[4]</code>	Value_64	Slave	Reset vector base address.
<code>clk_in</code>	<code>ClockSignal</code>	Slave	The clock signal connected to the <code>clk_in</code> port is used to determine the rate at which the core executes instructions.

Name	Protocol	Type	Description
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.
commt[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.
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.

Name	Protocol	Type	Description
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
topreset	Signal	Slave	This signal resets timer and interrupt controller.
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.
warmrstreq[4]	Signal	Master	Warm reset request from core.

Parameters for ARM_Cortex-R52Plus

cpu0.ase-present

Set whether the model has been built with NEON support.

Type: bool. Default value: 0x1.

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CFGTE

Equivalent to CFGTHUMBEXCEPTIONS.

Type: bool. Default value: 0x0.

cpu0.dcache-size

L1 D-Cache size in bytes.

Type: int. Default value: 0x8000.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.flash.enable

Equivalent to CFGFLASHEN.

Type: bool. Default value: 0x0.

cpu0.icache-size

L1 I-Cache size in bytes.

Type: int. Default value: 0x8000.

cpu0.llpp.base

Equivalent to CFGLLPPBASEADDR.

Type: int. Default value: 0x0.

cpu0.llpp.size

Equivalent to CFGLLPPSIZE.

Type: int. Default value: 0x8000000.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Equivalent to CFGVECTABLE.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: `int`. Default value: `0xab`.

cpu0.tcm.a.enable

Equivalent to CFGTCMBOOT.
Type: `bool`. Default value: `0x0`.

cpu0.tcm.a.size

Sets the size of the ATCM(in bytes).
Type: `int`. Default value: `0x4000`.

cpu0.tcm.a.wait

TCM Register A accesses wait states: 0-1 states.
Type: `int`. Default value: `0x0`.

cpu0.tcm.b.size

Sets the size of the BTCM(in bytes).
Type: `int`. Default value: `0x4000`.

cpu0.tcm.b.wait

TCM Register B accesses wait states: 0-1 states.
Type: `int`. Default value: `0x0`.

cpu0.tcm.c.size

Sets the size of the CTCM(in bytes).
Type: `int`. Default value: `0x2000`.

cpu0.tcm.c.wait

TCM Register C accesses wait states: 0-1 states.
Type: `int`. Default value: `0x0`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.
Type: `int`. Default value: `0xf000`.

cpu0.vfp-dp-present

Whether double-precision floating point feature is implemented (FEAT_F64MM).
Type: `bool`. Default value: `0x1`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-R52Plus**CLUSTER_ID**

CLUSTER_ID[15:8] equivalent to CFGMPIDRAFF2, CLUSTER_ID[7:0] equivalent to CFGMPIDRAFF1.
Type: `int`. Default value: `0x0`.

cluster_utid

Equivalent to CFGCLUSTERUTID.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

DBGROMADDR

Equivalent to CFGDBGROMADDR.

Type: `int`. Default value: `0x0`.

DBGROMADDRV

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

Type: `bool`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

enable_lock_step

(equivalent to `CFGSLSPPLIT`).

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

flash_protection_enable_at_reset

Equivalent to `CFGFLASHPROTEN`.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x0`.

has_export_m_port

The interrupt distributor has an optional interrupt export port for routing interrupts to an external device.

Type: `bool`. Default value: `0x1`.

has_flash

Equivalent to `CFGFLASHIMP`.

Type: `bool`. Default value: `0x0`.

has_flash_protection

Equivalent to `CFGFLASHPROTIMP`.

Type: `bool`. Default value: `0x1`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

memory.ext_slave_base

Equivalent to CFGAXISTCMBASEADDR.
Type: `int`. Default value: `0x0`.

memory.flash_base

Equivalent to CFGFLASHBASEADDR.
Type: `int`. Default value: `0x0`.

NUM_CORES

Number of cores in cluster.
Type: `int`. Default value: `0x1`.

num_protection_regions_s1

Number of v8-R protection regions.
Type: `int`. Default value: `0x18`.

num_protection_regions_s2

Number of v8-R hyp protection regions.
Type: `int`. Default value: `0x18`.

num_spi

Number of interrupts (SPI) into the internal GIC controller.
Type: `int`. Default value: `0x3c0`.

PERIPHBASE

Equivalent to CFGPERIPHBASE.
Type: `int`. Default value: `0x13080000`.

ram_protection_enable_at_reset

Equivalent to CFGGRAMPROTEN.
Type: `bool`. Default value: `0x0`.

use_architectural_names

Use names SP/LR/PC instead of R13/R14/R15.
Type: `bool`. Default value: `0x0`.

Parameters for gic_iri**gic_iri.A3-affinity-supported**

Device supports affinity level 3 values that are non-zero.
Type: `bool`. Default value: `0x0`.

gic_iri.add-output-cpu-wake-request-signal-from-redistributor

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.
Type: `bool`. Default value: `0x0`.

gic_iri.allow-LPIEN-clear

Allow RW behaviour on GICR_CTLR.LPIEN instead of set once.
Type: `bool`. Default value: `0x0`.

gic_iri.ARE-fixed-to-one

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

Type: `bool`. Default value: `0x0`.

`gic_iri.chip-count`

The total number of chips supported.

Type: `int`. Default value: `0x10`.

`gic_iri.chip-id`

Chip ID when multichip operation is enabled.

Type: `int`. Default value: `0x0`.

`gic_iri.clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged`

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`. Default value: `0x0`.

`gic_iri.common-lpi-configuration`

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).

Type: `int`. Default value: `0x0`.

`gic_iri.common-vPE-table-affinity`

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.

Type: `string`. Default value: `""`.

`gic_iri consolidators`

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.

Type: `string`. Default value: `""`.

`gic_iri.CPU-affinities`

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.

Type: `string`. Default value: `""`.

`gic_iri.CPU-affinities-file`

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.

Type: `string`. Default value: `""`.

`gic_iri.cross-chip-AMBA-is-ACE`

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.

Type: `bool`. Default value: `0x0`.

`gic_iri.delay-ITS-accesses`

Delay accesses from the ITS until GICR_SYNCRR is read.

Type: `bool`. Default value: `0x1`.

`gic_iri.delay-redistributor-accesses`

Delay memory accesses from the redistributor until GICR_SYNCRR is read.

Type: `bool`. Default value: `0x1`.

`gic_iri.direct-lpi-support`

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: `0x0`.

`gic_iri.DPG-ARE-only`

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

Type: `bool`. Default value: `0x0`.

`gic_iri.DPG-bits-implemented`

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

Type: `bool`. Default value: `0x0`.

`gic_iri.DS-fixed-to-zero`

Enable/disable support of single security state.

Type: `bool`. Default value: `0x0`.

`gic_iri.enable-local-cross-chip-addressing`

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.

Type: `bool`. Default value: `0x0`.

`gic_iri.enable-multichip-operation`

Enables multi-chip operation between Distributors in distributed GIC IRI.

Type: `bool`. Default value: `0x0`.

`gic_iri.enable_protocol_checking`

Enable/disable protocol checking at cpu interface.

Type: `bool`. Default value: `0x0`.

`gic_iri.enabled`

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

`gic_iri.extended-ppi-count`

Number of extended PPI supported.

Type: `int`. Default value: `0x0`.

`gic_iri.extended-spi-count`

Number of extended SPI supported.

Type: `int`. Default value: `0x0`.

gic_iri.fixed-routed-spis

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.

Type: string. Default value: "".

gic_iri.GICD-alias

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).

Type: int. Default value: 0x0.

gic_iri.GICD-legacy-registers-as-reserved

When ARE is RAO/WI, makes superfluous registers in GICD reserved (including for the purpose of STATUSR updates).

Type: bool. Default value: 0x0.

gic_iri.GICD_CTLR-DS-1-means-secure-only

If GICD_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: bool. Default value: 0x0.

gic_iri.GICD_ITARGETSR-RAZWI

If true, the GICD_ITARGETS registers are RAZ/WI.

Type: bool. Default value: 0x0.

gic_iri.GICD_PIDR

The value for the GICD_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: int. Default value: 0x0.

gic_iri.GICR-clear-enable-supported

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.

Type: bool. Default value: 0x0.

gic_iri.GICR-invalidate-registers-implemented

When true, the registers GICR_INVLPIR, GICR_INVALLR and GICR_SYNCRR are implemented.

Type: bool. Default value: 0x0.

gic_iri.GICR_PIDR

The value for the GICR_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: int. Default value: 0x0.

gic_iri.GICR_PROPBASER-read-only

GICR_PROPBASER register is read-only.

Type: bool. Default value: 0x0.

gic_iri.GICR_PROPBASER-reset-value

Value of GICR_PROPBASER on reset.

Type: int. Default value: 0x0.

gic_iri.gicv2-only

If true, when using the GICv3 model, pretend to be a GICv2 system.

Type: `bool`. Default value: `0x0`.

gic_iri.GITS_BASER0-entry-bytes

Number of bytes required per entry for GITS_BASER0 register.

Type: `int`. Default value: `0x8`.

gic_iri.GITS_BASER0-indirect-RAZ

Indirect field for GITS_BASER0 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

gic_iri.GITS_BASER0-type

Type field for GITS_BASER0 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

gic_iri.GITS_BASER1-entry-bytes

Number of bytes required per entry for GITS_BASER1 register.

Type: `int`. Default value: `0x8`.

gic_iri.GITS_BASER1-indirect-RAZ

Indirect field for GITS_BASER1 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

gic_iri.GITS_BASER1-type

Type field for GITS_BASER1 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

gic_iri.GITS_BASER2-entry-bytes

Number of bytes required per entry for GITS_BASER2 register.

Type: `int`. Default value: `0x8`.

gic_iri.GITS_BASER2-indirect-RAZ

Indirect field for GITS_BASER2 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

gic_iri.GITS_BASER2-type

Type field for GITS_BASER2 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

gic_iri.GITS_BASER3-entry-bytes

Number of bytes required per entry for GITS_BASER3 register.

Type: `int`. Default value: `0x8`.

gic_iri.GITS_BASER3-indirect-RAZ

Indirect field for GITS_BASER3 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

gic_iri.GITS_BASER3-type

Type field for GITS_BASER3 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

gic_iri.GITS_BASER4-entry-bytes

Number of bytes required per entry for GITS_BASER4 register.

Type: `int`. Default value: `0x8`.

gic_iri.GITS_BASER4-indirect-RAZ

Indirect field for GITS_BASER4 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

gic_iri.GITS_BASER4-type

Type field for GITS_BASER4 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

gic_iri.GITS_BASER5-entry-bytes

Number of bytes required per entry for GITS_BASER5 register.

Type: `int`. Default value: `0x8`.

gic_iri.GITS_BASER5-indirect-RAZ

Indirect field for GITS_BASER5 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

gic_iri.GITS_BASER5-type

Type field for GITS_BASER5 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

gic_iri.GITS_BASER6-entry-bytes

Number of bytes required per entry for GITS_BASER6 register.

Type: `int`. Default value: `0x8`.

gic_iri.GITS_BASER6-indirect-RAZ

Indirect field for GITS_BASER6 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

gic_iri.GITS_BASER6-type

Type field for GITS_BASER6 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

gic_iri.GITS_BASER7-entry-bytes

Number of bytes required per entry for GITS_BASER7 register.

Type: `int`. Default value: `0x8`.

gic_iri.GITS_BASER7-indirect-RAZ

Indirect field for GITS_BASER7 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

gic_iri.GITS_BASER7-type

Type field for GITS_BASER7 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: int. Default value: 0x0.

gic_iri.GITS_PIDR

The value for the GITS_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: int. Default value: 0x0.

gic_iri.has-two-security-states

If true, has two security states.

Type: bool. Default value: 0x1.

gic_iri.has_mpam

Enable MPAM support on ITS and RDs.

Type: bool. Default value: 0x0.

gic_iri.has_VPENDBASER-dirty-flag-on-load

GICR_VPENDBASER.Dirty reflects transient loading state when valid=1.

Type: bool. Default value: 0x0.

gic_iri.ICFGR-PPI-mask

Mask for writes to ICFGR registers that configure PPIs.

Type: int. Default value: 0xaaaaaaaa.

gic_iri.ICFGR-PPI-reset

Reset value for ICFGR registers that configure PPIs.

Type: int. Default value: 0x0.

gic_iri.ICFGR-rsvd-bit

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGRn for n>0.

Type: bool. Default value: 0x0.

gic_iri.ICFGR-SGI-mask

Mask for writes to ICFGR registers that configure SGIs.

Type: int. Default value: 0x0.

gic_iri.ICFGR-SGI-reset

Reset value for ICFGR registers that configure SGIs.

Type: int. Default value: 0xaaaaaaaa.

gic_iri.ICFGR-SPI-mask

Mask for writes to ICFGR registers that configure SPIs.

Type: int. Default value: 0xaaaaaaaa.

gic_iri.ICFGR-SPI-reset

Reset value for ICFGR registers that configure SPIs.

Type: int. Default value: 0x0.

gic_iri.ignore-generate-sgi-when-no-are

Ignore GenerateSgi packets coming from the CPU interface if both ARE_S and ARE_NS are 0.

Type: `bool`. Default value: `0x0`.

`gic_iri.IGROUP-PPI-mask`

Mask for writes to PPI bits in IGROUP registers.

Type: `int`. Default value: `0xffff`.

`gic_iri.IGROUP-PPI-reset`

Reset value for SGI bits in IGROUP registers.

Type: `int`. Default value: `0x0`.

`gic_iri.IGROUP-SGI-mask`

Mask for writes to SGI bits in IGROUP registers.

Type: `int`. Default value: `0xffff`.

`gic_iri.IGROUP-SGI-reset`

Reset value for SGI bits in IGROUP registers.

Type: `int`. Default value: `0x0`.

`gic_iri.IIDR`

GICD_IIDR and GICR_IIDR value.

Type: `int`. Default value: `0x0`.

`gic_iri.IRI-ID-bits`

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.

Type: `int`. Default value: `0x10`.

`gic_iri.irouter-default-mask`

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

Type: `string`. Default value: `""`.

`gic_iri.irouter-default-reset`

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

Type: `string`. Default value: `""`.

`gic_iri.IROUTER-IRM-RAZ-WI`

GICD_IROUTERn.InterruptRoutingMode is RAZ/WI.

Type: `bool`. Default value: `0x0`.

`gic_iri.irouter-mask-values`

Mask Value of IROUTER[n] register in the form 'n=a.b.c.d'. n can be ≥ 32 and ≤ 1019 .

Type: `string`. Default value: `""`.

`gic_iri.irouter-reset-values`

Reset Value of IROUTER[n] register in the form 'n=a.b.c.d or n=*'. n can be ≥ 32 and ≤ 1019 .

Type: `string`. Default value: `""`.

`gic_iri.ITS-BASER-force-page-alignment`

Force alignment of address written to a GITS_BASER register to the page size configured.

Type: `bool`. Default value: `0x1`.

gic_iri.ITS-cache-invalidate-on-disable

Sets the RO bit GITS_TYPER.INV. When true, after the following sequence: 1) GITS_CTLR.Enabled written 1-->0, 2) GITS_CTLR.Quiescent observed as 1, 3) GITS_BASER<n>.Valid written 1-->0, there is no cached information from the ITS memory structure pointed to by GITS_BASER<n>.

Type: `bool`. Default value: `0x0`.

gic_iri.ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x0`.

gic_iri.ITS-count

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

Type: `int`. Default value: `0x0`.

gic_iri.ITS-cumulative-collection-tables

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.

Type: `bool`. Default value: `0x1`.

gic_iri.ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

gic_iri.ITS-entry-size

Number of bytes required to store each entry in the ITT tables.

Type: `int`. Default value: `0x8`.

gic_iri.ITS-hardware-collection-count

Number of hardware collections held exclusively in the ITS.

Type: `int`. Default value: `0x0`.

gic_iri.ITS-ID-bits

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

gic_iri.ITS-legacy-iidr-typer-offset

Put the GITS_IIDR and GITS_TYPER registers at their older offset of 0x8 and 0x4 respectively.

Type: `bool`. Default value: `0x0`.

gic_iri.ITS-MOVALL-update-collections

Whether MOVALL command updates the collection entires.

Type: `bool`. Default value: `0x0`.

gic_iri.ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

Type: `int`. Default value: `0x0`.

gic_iri.ITS-threaded-command-queue

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: bool. Default value: 0x1.

gic_iri.ITS-TRANSLATE64R

Add an implementation specific register at 0x10008 supporting 64 bit TRANSLATER (dev[63:32], interrupt[31:0]).

Type: bool. Default value: 0x0.

gic_iri.ITS-use-physical-target-addresses

Use physical hardware addresses for targets in ITS commands -- must be true for distributed implementations.

Type: bool. Default value: 0x1.

gic_iri.ITS-vmovp-bit

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.

Type: bool. Default value: 0x0.

gic_iri.ITS0-base

Register base address for ITS0 (automatic if 0).

Type: int. Default value: 0x0.

gic_iri.ITS1-base

Register base address for ITS1 (automatic if 0).

Type: int. Default value: 0x0.

gic_iri.ITS2-base

Register base address for ITS2 (automatic if 0).

Type: int. Default value: 0x0.

gic_iri.ITS3-base

Register base address for ITS3 (automatic if 0).

Type: int. Default value: 0x0.

gic_iri.legacy-sgi-enable-rao

Enables for SGI associated with an ARE=0 regime are RAO/WI.

Type: bool. Default value: 0x0.

gic_iri.local-SEIs

Generate SEI to signal internal issues.

Type: bool. Default value: 0x0.

gic_iri.local-VSEIs

Generate VSEI to signal internal issues.

Type: bool. Default value: 0x0.

gic_iri.lockable-SPI-count

Number of SPIs that are locked down when CFGSDISABLE signal is asserted. Only applies for GICv2.

Type: int. Default value: 0x0.

gic_iri.LPI-cache-check-data

Enable Cached LPI data against memory checking when available for cache type.

Type: `bool`. Default value: `0x0`.

gic_iri.LPI-cache-type

Cache type for LPIs, 0:No caching, 1:Full caching.

Type: `int`. Default value: `0x1`.

gic_iri.max-pe-on-chip

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

gic_iri.monolithic

Indicate that the implementation is not distributed.

Type: `bool`. Default value: `0x0`.

gic_iri.mpam_max_partid

Maximum valid PARTID.

Type: `int`. Default value: `0xffff`.

gic_iri.mpam_max_pmg

Maximum valid PMG.

Type: `int`. Default value: `0xff`.

gic_iri.MSI_IIDR

Value returned in MSI_IIDR registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame0-base

If non-zero, sets the base address used for non-secure MSI frame 0 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame0-max-SPI

Maximum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame0-min-SPI

Minimum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame1-base

If non-zero, sets the base address used for non-secure MSI frame 1 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame1-max-SPI

Maximum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame1-min-SPI

Minimum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame2-base

If non-zero, sets the base address used for non-secure MSI frame 2 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame2-max-SPI

Maximum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame2-min-SPI

Minimum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame3-base

If non-zero, sets the base address used for non-secure MSI frame 3 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame3-max-SPI

Maximum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame3-min-SPI

Minimum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame4-base

If non-zero, sets the base address used for non-secure MSI frame 4 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame4-max-SPI

Maximum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame4-min-SPI

Minimum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame5-base

If non-zero, sets the base address used for non-secure MSI frame 5 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame5-max-SPI

Maximum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame5-min-SPI

Minimum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame6-base

If non-zero, sets the base address used for non-secure MSI frame 6 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame6-max-SPI

Maximum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame6-min-SPI

Minimum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame7-base

If non-zero, sets the base address used for non-secure MSI frame 7 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame7-max-SPI

Maximum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame7-min-SPI

Minimum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_PIDR

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.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame0-base

If non-zero, sets the base address used for secure MSI frame 0 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame0-max-SPI

Maximum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame0-min-SPI

Minimum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame1-base

If non-zero, sets the base address used for secure MSI frame 1 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame1-max-SPI

Maximum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame1-min-SPI

Minimum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame2-base

If non-zero, sets the base address used for secure MSI frame 2 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame2-max-SPI

Maximum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame2-min-SPI

Minimum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame3-base

If non-zero, sets the base address used for secure MSI frame 3 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame3-max-SPI

Maximum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame3-min-SPI

Minimum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame4-base

If non-zero, sets the base address used for secure MSI frame 4 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame4-max-SPI

Maximum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame4-min-SPI

Minimum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame5-base

If non-zero, sets the base address used for secure MSI frame 5 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame5-max-SPI

Maximum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame5-min-SPI

Minimum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame6-base

If non-zero, sets the base address used for secure MSI frame 6 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame6-max-SPI

Maximum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame6-min-SPI

Minimum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame7-base

If non-zero, sets the base address used for secure MSI frame 7 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame7-max-SPI

Maximum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame7-min-SPI

Minimum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if `enable-multichip-operation` is set.

Type: `bool`. Default value: `0x1`.

gic_iri.non-ARE-core-count

Maximum number of non-ARE cores; normally used to pass the cluster-level `NUM_CORES` parameter to the top-level redistributor.

Type: `int`. Default value: `0x8`.

gic_iri.outer-cacheability-support

Allow configuration of outer cacheability attributes in ITS and Redistributor.

Type: `bool`. Default value: `0x0`.

gic_iri.output_attributes

User-defined transform to be applied to bus attributes like `MasterID`, `ExtendedID` or `UserFlags`. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`. Default value: `"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS"`.

gic_iri.PA_SIZE

Number of valid bits in physical address.

Type: `int`. Default value: `0x30`.

gic_iri.PPI-implemented-mask

Mask of PPIs that are implemented. One bit per PPI bit 0 == PPI 16 (first PPI). This will affect other masks.

Type: `int`. Default value: `0xfffff`.

gic_iri.print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

gic_iri.priority-bits

Number of implemented priority bits.

Type: `int`. Default value: `0x5`.

gic_iri.processor-numbers

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.

Type: string. Default value: "".

gic_iri.redistributor-threaded-sync

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: bool. Default value: 0x1.

gic_iri.reg-base

Base for decoding GICv3 registers.

Type: int. Default value: 0x2c010000.

gic_iri.reg-base-per-redistributor

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.

Type: string. Default value: "".

gic_iri.reg-base-per-redistributor-file

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.

Type: string. Default value: "".

gic_iri.report-MSI-error-via-statusr

Report MSI error via GITS_STATUSR. (0:unsupported, 1:report by GITS_STATUSR, 2:report by GITS_STATUSR and interrupt as well).

Type: int. Default value: 0x0.

gic_iri.sgi-range-selector-support

Device has support for the Range Selector feature for SGI.

Type: bool. Default value: 0x0.

gic_iri.single-set-support

When true, forces redistributors to recall interrupts with a clear rather than issue a second Set command.

Type: bool. Default value: 0x0.

gic_iri.SPI-count

Number of SPIs that are implemented.

Type: int. Default value: 0xe0.

gic_iri.SPI-message-based-support

Distributor supports message based signaling of SPI.

Type: bool. Default value: 0x1.

`gic_iri.SPI-unimplemented`

A comma separated list of unimplemented SPIs ranges for sparse SPI definition(for ex: '35, 39-42, 73').

Type: `string`. Default value: `""`.

`gic_iri.STATUSR-implemented`

Determines whether the GICR_STATUSR register is implemented.

Type: `bool`. Default value: `0x1`.

`gic_iri.supports-shareability`

Device supports shareability attributes on outgoing memory bus (i.e. is modelling an ACElite port rather than an AXI4 port).

Type: `bool`. Default value: `0x1`.

`gic_iri.trace-speculative-lpi-property-update`

Trace LPI property updates performed on speculative accesses (useful for debugging LPI).

Type: `bool`. Default value: `0x0`.

`gic_iri.virtual-lpi-support`

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.

Type: `bool`. Default value: `0x0`.

`gic_iri.virtual-priority-bits`

Number of implemented virtual priority bits.

Type: `int`. Default value: `0x5`.

`gic_iri.wakeup-on-reset`

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.

Type: `bool`. Default value: `0x0`.

4.5.59 ARMCortexR52x1CT

ARMCortexR52x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-196: IP revisions support

Revision	Quality level
r1p4	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexR52x1CT contains the following CADI targets:

- `ARM_CortexR52`
- `Cluster_ARM_Cortex-R52`

- PVCache
- TlbCadi
- gic_iri

ARMCortexR52x1CT contains the following MTI components:

- [ARM_CortexR52](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [GICv3CPUInterface](#)
- [GICv3CPUInterfaceDecoder](#)
- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `CMO_broadcast_when_cache_state_modelling_disabled`
- `enable_lock_step`
- `enable_simulation_performance_optimizations`

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.

Ports for ARMCortexR52x1CT

Table 4-197: Ports

Name	Protocol	Type	Description
<code>cfgdbgromaddr</code>	Value_64	Slave	Debug ROM base address.
<code>cfgdbgromaddrv</code>	Signal	Slave	Debug ROM base address valid.
<code>cfgendianess[1]</code>	Signal	Slave	This signal if for EE bit initialisation.
<code>cfgperiphbase</code>	Value_64	Slave	This port sets the base address of private peripheral region.
<code>cfgthumbexceptions[1]</code>	Signal	Slave	This signal provides default exception handling state.
<code>cfgvectable[1]</code>	Value_64	Slave	Reset vector base address.
<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	Configure Aff2 and Aff1 fields of MPIDR. Aff1 = value[7:0], Aff2 = value[15:8]
<code>cntvalueb</code>	CounterInterface	Slave	Interface to SoC level counter module.
<code>commr[1]</code>	Signal	Master	Receive portion of Data Transfer Register full.
<code>commtx[1]</code>	Signal	Master	Transmit portion of Data Transfer Register empty.
<code>cpuhalt[1]</code>	Signal	Slave	Raising this signal will put the core into halt mode.
<code>cpuporeset[1]</code>	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
<code>cti[1]</code>	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
<code>dbgack[1]</code>	Signal	Master	External debug interface.

Name	Protocol	Type	Description
dbggen[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
dbgpwrupreq[1]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
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 ARM_CortexR52

cpu0.ase-present

Set whether the model has been built with NEON support.

Type: bool. Default value: 0x1.

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: `bool`. Default value: `0x0`.

cpu0.CFGTE

Equivalent to `CFGTHUMBEXCEPTIONS`.
Type: `bool`. Default value: `0x0`.

cpu0.dcache-size

L1 D-Cache size in bytes.
Type: `int`. Default value: `0x8000`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.
Type: `bool`. Default value: `0x0`.

cpu0.flash.enable

Equivalent to `CFGFLASHEN`.
Type: `bool`. Default value: `0x0`.

cpu0.icache-size

L1 I-Cache size in bytes.
Type: `int`. Default value: `0x8000`.

cpu0.llpp.base

Equivalent to `CFGLLPPBASEADDR`.
Type: `int`. Default value: `0x0`.

cpu0.llpp.size

Equivalent to `CFGLLPPSIZE`.
Type: `int`. Default value: `0x1000`.

cpu0.max_code_cache_mb

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).
Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

Force minimum `syncLevel` (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll).
Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Equivalent to `CFGVECTABLE`.
Type: `int`. Default value: `0x0`.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.
Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.
Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.
Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.
Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xffffffff.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: int. Default value: 0xffffffff.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: int. Default value: 0x3c.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: int. Default value: 0xab.

cpu0.tcm.a.enable

Equivalent to CFGTCMBOOT.
Type: bool. Default value: 0x0.

cpu0.tcm.a.size

Sets the size of the ATCM(in bytes).
Type: int. Default value: 0x4000.

cpu0.tcm.a.wait

TCM Register A accesses wait states: 0-1 states.
Type: int. Default value: 0x0.

cpu0.tcm.b.size

Sets the size of the BTCM(in bytes).
Type: int. Default value: 0x4000.

cpu0.tcm.b.wait

TCM Register B accesses wait states: 0-1 states.

Type: `int`. Default value: `0x0`.

cpu0.tcm.c.size

Sets the size of the CTCM(in bytes).

Type: `int`. Default value: `0x2000`.

cpu0.tcm.c.wait

TCM Register C accesses wait states: 0-1 states.

Type: `int`. Default value: `0x0`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-dp-present

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

Type: `bool`. Default value: `0x1`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-R52

CLUSTER_ID

CLUSTER_ID[15:8] equivalent to CFGMPIDRAFF2, CLUSTER_ID[7:0] equivalent to CFGMPIDRAFF1.

Type: `int`. Default value: `0x0`.

cluster_utid

Equivalent to CFGCLUSTERUTID.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

DBGROMADDR

Equivalent to `CFGDBGROMADDR`.

Type: `int`. Default value: `0x0`.

DBGROMADDRV

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

Type: `bool`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

enable_lock_step

(equivalent to `CFGSLSPPLIT`).

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

flash_protection_enable_at_reset

Equivalent to `CFGFLASHPROTEN`.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x0`.

has_export_m_port

The interrupt distributor has an optional interrupt export port for routing interrupts to an external device.

Type: `bool`. Default value: `0x1`.

has_flash

Equivalent to `CFGFLASHIMP`.

Type: `bool`. Default value: `0x0`.

has_flash_protection

Equivalent to `CFGFLASHPROTIMP`.

Type: `bool`. Default value: `0x1`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

memory.ext_slave_base

Equivalent to `CFGAXISTCMBASEADDR`.

Type: `int`. Default value: `0x0`.

memory.flash_base

Equivalent to `CFGFLASHBASEADDR`.

Type: `int`. Default value: `0x0`.

num_protection_regions_s1

Number of v8-R protection regions.

Type: `int`. Default value: `0x18`.

num_protection_regions_s2

Number of v8-R hyp protection regions.

Type: `int`. Default value: `0x18`.

num_spi

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

Type: `int`. Default value: `0x3c0`.

PERIPHBASE

Equivalent to `CFGPERIPHBASE`.

Type: `int`. Default value: `0x13080000`.

ram_protection_enable_at_reset

Equivalent to `CFGGRAMPROTEN`.

Type: `bool`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

Parameters for `gic_iri`

`gic_iri.A3-affinity-supported`

Device supports affinity level 3 values that are non-zero.

Type: `bool`. Default value: `0x0`.

`gic_iri.add-output-cpu-wake-request-signal-from-redistributor`

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.

Type: `bool`. Default value: `0x0`.

`gic_iri.allow-LPIEN-clear`

Allow RW behaviour on `GICR_CTLR.LPIEN` instead of set once.

Type: `bool`. Default value: `0x0`.

`gic_iri.ARE-fixed-to-one`

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

Type: `bool`. Default value: `0x0`.

`gic_iri.chip-count`

The total number of chips supported.

Type: `int`. Default value: `0x10`.

`gic_iri.chip-id`

Chip ID when multichip operation is enabled.

Type: `int`. Default value: `0x0`.

`gic_iri.clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged`

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the `ISPENDR` register.

Type: `bool`. Default value: `0x0`.

`gic_iri.common-lpi-configuration`

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).

Type: `int`. Default value: `0x0`.

`gic_iri.common-vPE-table-affinity`

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.

Type: `string`. Default value: `""`.

`gic_iri.consolidators`

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.

Type: `string`. Default value: `""`.

`gic_iri.CPU-affinities`

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.

Type: `string`. Default value: `""`.

`gic_iri.CPU-affinities-file`

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.

Type: `string`. Default value: `""`.

`gic_iri.cross-chip-AMBA-is-ACE`

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.

Type: `bool`. Default value: `0x0`.

`gic_iri.delay-ITS-accesses`

Delay accesses from the ITS until GICR_SYNCRR is read.

Type: `bool`. Default value: `0x1`.

`gic_iri.delay-redistributor-accesses`

Delay memory accesses from the redistributor until GICR_SYNCRR is read.

Type: `bool`. Default value: `0x1`.

`gic_iri.direct-lpi-support`

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: `0x0`.

`gic_iri.DPG-ARE-only`

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

Type: `bool`. Default value: `0x0`.

`gic_iri.DPG-bits-implemented`

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

Type: `bool`. Default value: `0x0`.

`gic_iri.DS-fixed-to-zero`

Enable/disable support of single security state.

Type: `bool`. Default value: `0x0`.

`gic_iri.enable-local-cross-chip-addressing`

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.

Type: `bool`. Default value: `0x0`.

`gic_iri.enable-multichip-operation`

Enables multi-chip operation between Distributors in distributed GIC IRI.

Type: `bool`. Default value: `0x0`.

`gic_iri.enable_protocol_checking`

Enable/disable protocol checking at cpu interface.

Type: `bool`. Default value: `0x0`.

`gic_iri.enabled`

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

`gic_iri.extended-ppi-count`

Number of extended PPI supported.

Type: `int`. Default value: `0x0`.

`gic_iri.extended-spi-count`

Number of extended SPI supported.

Type: `int`. Default value: `0x0`.

`gic_iri.fixed-routed-spis`

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 .

Type: `string`. Default value: `""`.

`gic_iri.GICD-alias`

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).

Type: `int`. Default value: `0x0`.

`gic_iri.GICD-legacy-registers-as-reserved`

When ARE is RAO/WI, makes superfluous registers in GICD reserved (including for the purpose of STATUSR updates).

Type: `bool`. Default value: `0x0`.

`gic_iri.GICD_CTLR-DS-1-means-secure-only`

If GICD_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: `bool`. Default value: `0x0`.

gic_iri.GICD_ITARGETSR-RAZWI

If true, the GICD_ITARGETS registers are RAZ/WI.

Type: `bool`. Default value: `0x0`.

gic_iri.GICD_PIDR

The value for the GICD_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `int`. Default value: `0x0`.

gic_iri.GICR-clear-enable-supported

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.

Type: `bool`. Default value: `0x0`.

gic_iri.GICR-invalidate-registers-implemented

When true, the registers GICR_INVLPIR, GICR_INVALLR and GICR_SYNCR are implemented.

Type: `bool`. Default value: `0x0`.

gic_iri.GICR_PIDR

The value for the GICR_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `int`. Default value: `0x0`.

gic_iri.GICR_PROPBASER-read-only

GICR_PROPBASER register is read-only.

Type: `bool`. Default value: `0x0`.

gic_iri.GICR_PROPBASER-reset-value

Value of GICR_PROPBASER on reset.

Type: `int`. Default value: `0x0`.

gic_iri.gicv2-only

If true, when using the GICv3 model, pretend to be a GICv2 system.

Type: `bool`. Default value: `0x0`.

gic_iri.GITS_BASER0-entry-bytes

Number of bytes required per entry for GITS_BASER0 register.

Type: `int`. Default value: `0x8`.

gic_iri.GITS_BASER0-indirect-RAZ

Indirect field for GITS_BASER0 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

gic_iri.GITS_BASER0-type

Type field for GITS_BASER0 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

gic_iri.GITS_BASER1-entry-bytes

Number of bytes required per entry for GITS_BASER1 register.

Type: `int`. Default value: `0x8`.

gic_iri.GITS_BASER1-indirect-RAZ

Indirect field for GITS_BASER1 register is RAZ/WI.

Type: bool. Default value: 0x0.

gic_iri.GITS_BASER1-type

Type field for GITS_BASER1 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: int. Default value: 0x0.

gic_iri.GITS_BASER2-entry-bytes

Number of bytes required per entry for GITS_BASER2 register.

Type: int. Default value: 0x8.

gic_iri.GITS_BASER2-indirect-RAZ

Indirect field for GITS_BASER2 register is RAZ/WI.

Type: bool. Default value: 0x0.

gic_iri.GITS_BASER2-type

Type field for GITS_BASER2 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: int. Default value: 0x0.

gic_iri.GITS_BASER3-entry-bytes

Number of bytes required per entry for GITS_BASER3 register.

Type: int. Default value: 0x8.

gic_iri.GITS_BASER3-indirect-RAZ

Indirect field for GITS_BASER3 register is RAZ/WI.

Type: bool. Default value: 0x0.

gic_iri.GITS_BASER3-type

Type field for GITS_BASER3 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: int. Default value: 0x0.

gic_iri.GITS_BASER4-entry-bytes

Number of bytes required per entry for GITS_BASER4 register.

Type: int. Default value: 0x8.

gic_iri.GITS_BASER4-indirect-RAZ

Indirect field for GITS_BASER4 register is RAZ/WI.

Type: bool. Default value: 0x0.

gic_iri.GITS_BASER4-type

Type field for GITS_BASER4 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: int. Default value: 0x0.

gic_iri.GITS_BASER5-entry-bytes

Number of bytes required per entry for GITS_BASER5 register.

Type: int. Default value: 0x8.

gic_iri.GITS_BASER5-indirect-RAZ

Indirect field for GITS_BASER5 register is RAZ/WI.

Type: bool. Default value: 0x0.

gic_iri.GITS_BASER5-type

Type field for GITS_BASER5 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: int. Default value: 0x0.

gic_iri.GITS_BASER6-entry-bytes

Number of bytes required per entry for GITS_BASER6 register.

Type: int. Default value: 0x8.

gic_iri.GITS_BASER6-indirect-RAZ

Indirect field for GITS_BASER6 register is RAZ/WI.

Type: bool. Default value: 0x0.

gic_iri.GITS_BASER6-type

Type field for GITS_BASER6 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: int. Default value: 0x0.

gic_iri.GITS_BASER7-entry-bytes

Number of bytes required per entry for GITS_BASER7 register.

Type: int. Default value: 0x8.

gic_iri.GITS_BASER7-indirect-RAZ

Indirect field for GITS_BASER7 register is RAZ/WI.

Type: bool. Default value: 0x0.

gic_iri.GITS_BASER7-type

Type field for GITS_BASER7 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: int. Default value: 0x0.

gic_iri.GITS_PIDR

The value for the GITS_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: int. Default value: 0x0.

gic_iri.has-two-security-states

If true, has two security states.

Type: bool. Default value: 0x1.

gic_iri.has_mpam

Enable MPAM support on ITS and RDs.

Type: bool. Default value: 0x0.

gic_iri.has_VPENDBASER-dirty-flag-on-load

GICR_VPENDBASER.Dirty reflects transient loading state when valid=1.

Type: bool. Default value: 0x0.

gic_iri.ICFGR-PPI-mask

Mask for writes to ICFGR registers that configure PPIs.

Type: int. Default value: 0xaaaaaaaa.

gic_iri.ICFGR-PPI-reset

Reset value for ICFGR registers that configure PPIs.

Type: int. Default value: 0x0.

gic_iri.ICFGR-rsvd-bit

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGRn for n>0.

Type: bool. Default value: 0x0.

gic_iri.ICFGR-SGI-mask

Mask for writes to ICFGR registers that configure SGIs.

Type: int. Default value: 0x0.

gic_iri.ICFGR-SGI-reset

Reset value for ICFGR registers that configure SGIs.

Type: int. Default value: 0xaaaaaaaa.

gic_iri.ICFGR-SPI-mask

Mask for writes to ICFGR registers that configure SPIs.

Type: int. Default value: 0xaaaaaaaa.

gic_iri.ICFGR-SPI-reset

Reset value for ICFGR registers that configure SPIs.

Type: int. Default value: 0x0.

gic_iri.ignore-generate-sgi-when-no-are

Ignore GenerateSGI packets coming from the CPU interface if both ARE_S and ARE_NS are 0.

Type: bool. Default value: 0x0.

gic_iri.IGROUP-PPI-mask

Mask for writes to PPI bits in IGROUP registers.

Type: int. Default value: 0xfffff.

gic_iri.IGROUP-PPI-reset

Reset value for SGI bits in IGROUP registers.

Type: int. Default value: 0x0.

gic_iri.IGROUP-SGI-mask

Mask for writes to SGI bits in IGROUP registers.

Type: int. Default value: 0xfffff.

gic_iri.IGROUP-SGI-reset

Reset value for SGI bits in IGROUP registers.

Type: int. Default value: 0x0.

gic_iri.IIDR

GICD_IIDR and GICR_IIDR value.

Type: int. Default value: 0x0.

gic_iri.IRI-ID-bits

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.

Type: `int`. Default value: `0x10`.

gic_iri.irouter-default-mask

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

Type: `string`. Default value: `""`.

gic_iri.irouter-default-reset

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

Type: `string`. Default value: `""`.

gic_iri.IROUTER-IRM-RAZ-WI

GICD_IROUTERn.InterruptRoutingMode is RAZ/WI.

Type: `bool`. Default value: `0x0`.

gic_iri.irouter-mask-values

Mask Value of IROUTER[n] register in the form 'n=a.b.c.d'. n can be ≥ 32 and ≤ 1019 .

Type: `string`. Default value: `""`.

gic_iri.irouter-reset-values

Reset Value of IROUTER[n] register in the form 'n=a.b.c.d' or 'n=*'. n can be ≥ 32 and ≤ 1019 .

Type: `string`. Default value: `""`.

gic_iri.ITS-BASER-force-page-alignment

Force alignment of address written to a GITS_BASER register to the page size configured.

Type: `bool`. Default value: `0x1`.

gic_iri.ITS-cache-invalidate-on-disable

Sets the RO bit GITS_TYPER.INV. When true, after the following sequence: 1) GITS_CTLR.Enabled written 1-->0, 2) GITS_CTLR.Quiescent observed as 1, 3) GITS_BASER<n>.Valid written 1-->0, there is no cached information from the ITS memory structure pointed to by GITS_BASER<n>.

Type: `bool`. Default value: `0x0`.

gic_iri.ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x0`.

gic_iri.ITS-count

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

Type: `int`. Default value: `0x0`.

gic_iri.ITS-cumulative-collection-tables

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.

Type: `bool`. Default value: `0x1`.

`gic_iri.ITS-device-bits`

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

`gic_iri.ITS-entry-size`

Number of bytes required to store each entry in the ITT tables.

Type: `int`. Default value: `0x8`.

`gic_iri.ITS-hardware-collection-count`

Number of hardware collections held exclusively in the ITS.

Type: `int`. Default value: `0x0`.

`gic_iri.ITS-ID-bits`

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

`gic_iri.ITS-legacy-iidr-typer-offset`

Put the GITS_IIDR and GITS_TYPER registers at their older offset of `0x8` and `0x4` respectively.

Type: `bool`. Default value: `0x0`.

`gic_iri.ITS-MOVALL-update-collections`

Whether MOVALL command updates the collection entires.

Type: `bool`. Default value: `0x0`.

`gic_iri.ITS-shared-vPE-table`

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `int`. Default value: `0x0`.

`gic_iri.ITS-threaded-command-queue`

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

`gic_iri.ITS-TRANSLATE64R`

Add an implementation specific register at `0x10008` supporting 64 bit TRANSLATER (`dev[63:32]`, `interrupt[31:0]`).

Type: `bool`. Default value: `0x0`.

`gic_iri.ITS-use-physical-target-addresses`

Use physical hardware addresses for targets in ITS commands -- must be true for distributed implementations.

Type: `bool`. Default value: `0x1`.

`gic_iri.ITS-vmovp-bit`

Device supports software issuing a VMOVPP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVPP command across all ITSs that have mapping for that vPE.

Type: `bool`. Default value: `0x0`.

`gic_iri.ITS0-base`

Register base address for ITS0 (automatic if 0).

Type: `int`. Default value: `0x0`.

`gic_iri.ITS1-base`

Register base address for ITS1 (automatic if 0).

Type: `int`. Default value: `0x0`.

`gic_iri.ITS2-base`

Register base address for ITS2 (automatic if 0).

Type: `int`. Default value: `0x0`.

`gic_iri.ITS3-base`

Register base address for ITS3 (automatic if 0).

Type: `int`. Default value: `0x0`.

`gic_iri.legacy-sgi-enable-rao`

Enables for SGI associated with an ARE=0 regime are RAO/WI.

Type: `bool`. Default value: `0x0`.

`gic_iri.local-SEIs`

Generate SEI to signal internal issues.

Type: `bool`. Default value: `0x0`.

`gic_iri.local-VSEIs`

Generate VSEI to signal internal issues.

Type: `bool`. Default value: `0x0`.

`gic_iri.lockable-SPI-count`

Number of SPIs that are locked down when CFGSDISABLE signal is asserted. Only applies for GICv2.

Type: `int`. Default value: `0x0`.

`gic_iri.LPI-cache-check-data`

Enable Cached LPI data against memory checking when available for cache type.

Type: `bool`. Default value: `0x0`.

`gic_iri.LPI-cache-type`

Cache type for LPIs, 0:No caching, 1:Full caching.

Type: `int`. Default value: `0x1`.

`gic_iri.max-pe-on-chip`

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

`gic_iri.monolithic`

Indicate that the implementation is not distributed.

Type: `bool`. Default value: `0x0`.

`gic_iri.mpam_max_partid`

Maximum valid PARTID.

Type: `int`. Default value: `0xffff`.

`gic_iri.mpam_max_pmg`

Maximum valid PMG.

Type: `int`. Default value: `0xff`.

`gic_iri.MSI_IIDR`

Value returned in MSI_IIDR registers.

Type: `int`. Default value: `0x0`.

`gic_iri.MSI_NS-frame0-base`

If non-zero, sets the base address used for non-secure MSI frame 0 registers.

Type: `int`. Default value: `0x0`.

`gic_iri.MSI_NS-frame0-max-SPI`

Maximum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

`gic_iri.MSI_NS-frame0-min-SPI`

Minimum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

`gic_iri.MSI_NS-frame1-base`

If non-zero, sets the base address used for non-secure MSI frame 1 registers.

Type: `int`. Default value: `0x0`.

`gic_iri.MSI_NS-frame1-max-SPI`

Maximum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

`gic_iri.MSI_NS-frame1-min-SPI`

Minimum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

`gic_iri.MSI_NS-frame2-base`

If non-zero, sets the base address used for non-secure MSI frame 2 registers.

Type: `int`. Default value: `0x0`.

`gic_iri.MSI_NS-frame2-max-SPI`

Maximum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

`gic_iri.MSI_NS-frame2-min-SPI`

Minimum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

`gic_iri.MSI_NS-frame3-base`

If non-zero, sets the base address used for non-secure MSI frame 3 registers.

Type: `int`. Default value: `0x0`.

`gic_iri.MSI_NS-frame3-max-SPI`

Maximum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

`gic_iri.MSI_NS-frame3-min-SPI`

Minimum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame4-base

If non-zero, sets the base address used for non-secure MSI frame 4 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame4-max-SPI

Maximum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame4-min-SPI

Minimum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame5-base

If non-zero, sets the base address used for non-secure MSI frame 5 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame5-max-SPI

Maximum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame5-min-SPI

Minimum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame6-base

If non-zero, sets the base address used for non-secure MSI frame 6 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame6-max-SPI

Maximum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame6-min-SPI

Minimum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame7-base

If non-zero, sets the base address used for non-secure MSI frame 7 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame7-max-SPI

Maximum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_NS-frame7-min-SPI

Minimum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_PIDR

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.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame0-base

If non-zero, sets the base address used for secure MSI frame 0 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame0-max-SPI

Maximum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame0-min-SPI

Minimum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame1-base

If non-zero, sets the base address used for secure MSI frame 1 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame1-max-SPI

Maximum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame1-min-SPI

Minimum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame2-base

If non-zero, sets the base address used for secure MSI frame 2 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame2-max-SPI

Maximum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame2-min-SPI

Minimum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame3-base

If non-zero, sets the base address used for secure MSI frame 3 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame3-max-SPI

Maximum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame3-min-SPI

Minimum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame4-base

If non-zero, sets the base address used for secure MSI frame 4 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame4-max-SPI

Maximum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame4-min-SPI

Minimum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame5-base

If non-zero, sets the base address used for secure MSI frame 5 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame5-max-SPI

Maximum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame5-min-SPI

Minimum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame6-base

If non-zero, sets the base address used for secure MSI frame 6 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame6-max-SPI

Maximum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame6-min-SPI

Minimum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame7-base

If non-zero, sets the base address used for secure MSI frame 7 registers.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame7-max-SPI

Maximum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.MSI_S-frame7-min-SPI

Minimum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

gic_iri.multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if `enable-multichip-operation` is set.

Type: `bool`. Default value: `0x1`.

gic_iri.non-ARE-core-count

Maximum number of non-ARE cores; normally used to pass the cluster-level `NUM_CORES` parameter to the top-level redistributor.

Type: `int`. Default value: `0x8`.

gic_iri.outer-cacheability-support

Allow configuration of outer cachability attributes in ITS and Redistributor.

Type: `bool`. Default value: `0x0`.

gic_iri.output_attributes

User-defined transform to be applied to bus attributes like MasterID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`. Default value: "ExtendedID[62:55]=MPAM_PMG,
ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS".

gic_iri.PA_SIZE

Number of valid bits in physical address.

Type: `int`. Default value: `0x30`.

gic_iri.PPI-implemented-mask

Mask of PPIs that are implemented. One bit per PPI bit 0 == PPI 16 (first PPI). This will affect other masks.

Type: `int`. Default value: `0xfffff`.

gic_iri.print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

gic_iri.priority-bits

Number of implemented priority bits.

Type: `int`. Default value: `0x5`.

gic_iri.processor-numbers

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.

Type: `string`. Default value: "".

gic_iri.redistributor-threaded-sync

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

gic_iri.reg-base

Base for decoding GICv3 registers.

Type: `int`. Default value: `0x2c010000`.

gic_iri.reg-base-per-redistributor

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.

Type: `string`. Default value: "".

gic_iri.reg-base-per-redistributor-file

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.

Type: `string`. Default value: `""`.

`gic_iri.report-MSI-error-via-statusr`

Report MSI error via GITS_STATUSR. (0:unsupported, 1:report by GITS_STATUSR, 2:report by GITS_STATUSR and interrupt as well).

Type: `int`. Default value: `0x0`.

`gic_iri.sgi-range-selector-support`

Device has support for the Range Selector feature for SGI.

Type: `bool`. Default value: `0x0`.

`gic_iri.single-set-support`

When true, forces redistributors to recall interrupts with a clear rather than issue a second Set command.

Type: `bool`. Default value: `0x0`.

`gic_iri.SPI-count`

Number of SPIs that are implemented.

Type: `int`. Default value: `0xe0`.

`gic_iri.SPI-message-based-support`

Distributor supports message based signaling of SPI.

Type: `bool`. Default value: `0x1`.

`gic_iri.SPI-unimplemented`

A comma separated list of unimplemented SPIs ranges for sparse SPI definition(for ex: '35, 39-42, 73').

Type: `string`. Default value: `""`.

`gic_iri.STATUSR-implemented`

Determines whether the GICR_STATUSR register is implemented.

Type: `bool`. Default value: `0x1`.

`gic_iri.supports-shareability`

Device supports shareability attributes on outgoing memory bus (i.e. is modelling an ACElite port rather than an AXI4 port).

Type: `bool`. Default value: `0x1`.

`gic_iri.trace-speculative-lpi-property-update`

Trace LPI property updates performed on speculative accesses (useful for debugging LPI).

Type: `bool`. Default value: `0x0`.

`gic_iri.virtual-lpi-support`

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.

Type: `bool`. Default value: `0x0`.

`gic_iri.virtual-priority-bits`

Number of implemented virtual priority bits.

Type: `int`. Default value: `0x5`.

gic_iri.wakeup-on-reset

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.

Type: `bool`. Default value: `0x0`.

4.5.60 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 4-198: IP revisions support

Revision	Quality level
r0p0	Preliminary support
r1p0	Preliminary support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM CortexR82CT contains the following CADI targets:

- ARM_Cortex-R82
- Cluster_ARM_Cortex-R82
- PVCache
- TlbCadi

ARM CortexR82CT contains the following MTI components:

- [ARM_Cortex-R82](#)
- [ARMv8Cluster](#)
- [DSU](#)
- [GICv3CPUInterface](#)
- [GICv3CPUInterfaceDecoder](#)
- [LLRAMAtomicOpFilter](#)
- [PPUv1](#)
- [PVBusExclusiveMonitor](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusMaster](#)
- [PVBusSlave](#)
- [PVCache](#)

- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

About ARMCortexR82CT

To simulate the r1p0 model, use the following parameters:

- `revision_number=1`
- `VMSA_supported=1`

The following models also exist:

- ARMCortexR82x1CT
- ARMCortexR82x2CT
- ARMCortexR82x3CT
- ARMCortexR82x4CT
- ARMCortexR82x6CT
- ARMCortexR82x8CT

Limitations

Dense memory map support has been added for the Utility bus only.

Ports for ARMCortexR82CT

Table 4-199: Ports

Name	Protocol	Type	Description
<code>acel_s</code>	PVBus	Slave	External Slave port. Equivalent to AXIS port.
<code>broadcastatomic</code>	Signal	Slave	CHI defined pins.
<code>broadcastatomicl</code>	Signal	Slave	BROADCASTATOMIC pin for LLRAM
<code>broadcastcachemaint</code>	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
<code>broadcastouter</code>	Signal	Slave	Enable broadcasting of Outer Shareable transactions.
<code>cache_validation_control</code>	Value	Slave	This signal provides default exception handling state.
<code>cfgendianess[8]</code>	Signal	Slave	This signal is for EE bit initialisation
<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>CNTHPSIRQ[8]</code>	Signal	Master	Timer signals to SOC

Name	Protocol	Type	Description
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.
coreerrirq[8]	Signal	Master	Core RAS error interrupt
corefaultirq[8]	Signal	Master	Core RAS fault interrupt
cp15sdisable[8]	Signal	Slave	This signal disables write access to some system control processor registers
cpuhalt[8]	Signal	Slave	Raising this signal will put the core into halt mode.
cti[8]	v8EmbeddedCrossTrigger_controlprotocol	Master	-
cti0extin[4]	Signal	Slave	CTI trace inputs for core 0.
cti0extout[4]	Signal	Master	CTI trace outputs for core 0.
cti1extin[4]	Signal	Slave	CTI trace inputs for core 1.
cti1extout[4]	Signal	Master	CTI trace outputs for core 1.
cti2extin[4]	Signal	Slave	CTI trace inputs for core 2.
cti2extout[4]	Signal	Master	CTI trace outputs for core 2.
cti3extin[4]	Signal	Slave	CTI trace inputs for core 3.
cti3extout[4]	Signal	Master	CTI trace outputs for core 3.
cti4extin[4]	Signal	Slave	CTI trace inputs for core 4.
cti4extout[4]	Signal	Master	CTI trace outputs for core 4.
cti5extin[4]	Signal	Slave	CTI trace inputs for core 5.
cti5extout[4]	Signal	Master	CTI trace outputs for core 5.
cti6extin[4]	Signal	Slave	CTI trace inputs for core 6.
cti6extout[4]	Signal	Master	CTI trace outputs for core 6.
cti7extin[4]	Signal	Slave	CTI trace inputs for core 7.
cti7extout[4]	Signal	Master	CTI trace outputs for core 7.
ctidbgirq[8]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	-
dbgnopwrdown[8]	Signal	Master	These signals relate to core power down.
dbgpwrdownack[8]	Signal	Master	Debug power down acknowledge.
dbgpwrdownreq[8]	Signal	Slave	Debug power down request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
external_trace_reset[8]	Signal	Slave	ETMv4 External Trace Reset signal.

Name	Protocol	Type	Description
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).
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.

Name	Protocol	Type	Description
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-R82

cpu0.ase-present

Set whether the model has been built with NEON support.

Type: bool. Default value: 0x1.

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CP15SDISABLE

Initialize to disable access to some CP15 registers.

Type: bool. Default value: 0x0.

cpu0.dcache-size

L1 D-Cache size in bytes.

Type: int. Default value: 0x10000.

cpu0.dtcn_base

Sets the 16K aligned base address of DTCM.

Type: int. Default value: 0x0.

cpu0.dtcn_size

Sets the size of DTCM (in bytes).

Type: int. Default value: 0x0.

cpu0.dtcn_stretch_clk

Whether DTCM clock stretched to occupy full cycle.

Type: bool. Default value: 0x0.

cpu0.dtcn_wait

DTCM accesses wait states: 0-3 cycles.

Type: int. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.icache-size

L1 I-Cache size in bytes.

Type: int. Default value: 0x10000.

cpu0.itcn_base

Sets the 16K aligned base address of ITCM.

Type: `int`. Default value: `0x0`.

`cpu0.itcm_size`

Sets the size of ITCM (in bytes).

Type: `int`. Default value: `0x0`.

`cpu0.itcm_stretch_clk`

Whether ITCM clock stretched to occupy full cycle.

Type: `bool`. Default value: `0x0`.

`cpu0.itcm_wait`

ITCM accesses wait states: 0-3 cycles.

Type: `int`. Default value: `0x0`.

`cpu0.llpp.base`

Equivalent to `CFGLLPPBASEADDR`.

Type: `int`. Default value: `0x0`.

`cpu0.llpp.size`

Equivalent to `CFGLLPPSIZE`.

Type: `int`. Default value: `0x8000000`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.RVBAR`

Value of `RVBAR_ELx` register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0x1000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0x1000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: int. Default value: 0x10000000.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: int. Default value: 0xf000000.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: int. Default value: 0x3c.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: int. Default value: 0xab.

cpu0.tcm.a.enable

Equivalent to CFGITCMENm.

Type: bool. Default value: 0x0.

cpu0.TEINIT

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

Type: bool. Default value: 0x0.

cpu0.trace_special_hlt_imm16

For this HLT number, IF enable_trace_special_hlt_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: int. Default value: 0xf000.

cpu0.vfp-dp-present

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

Type: bool. Default value: 0x1.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool. Default value: 0x0.

cpu0.vfp-present

Set whether the model has VFP support.

Type: bool. Default value: 0x1.

Parameters for Cluster_ARM_Cortex-R82

BROADCASTATOMIC

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

Type: bool. Default value: 0x1.

BROADCASTATOMICL

Enable broadcasting of atomic operation to LLRAM memory. The broadcastatomicl signal will override this value and it will be functional for revision 2.

Type: bool. Default value: 0x1.

BROADCASTCACHEMAINT

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

Type: bool. Default value: 0x1.

BROADCASTOUTER

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

Type: bool. Default value: 0x1.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: int. Default value: 0x0.

CCSIDR-L1D_override

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

Type: int. Default value: 0x0.

CCSIDR-L1I_override

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

Type: int. Default value: 0x0.

CCSIDR-L2_override

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

Type: int. Default value: 0x0.

CCSIDR-L3_override

If nonzero, allow L3 selection in CSSELR and present this value in CCSIDR (this is cosmetic and does not affect cache behaviour).

Type: int. Default value: 0x0.

CLUSTER_ID

CLUSTER_ID[15:8] equivalent to CFGMPIDRAFF3, CLUSTER_ID[7:0] equivalent to CFGMPIDRAFF2.

Type: int. Default value: 0x0.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

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.

Type: `bool`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

DBGROMADDR

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

Type: `int`. Default value: `0x0`.

DBGROMADDRV

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

Type: `bool`. Default value: `0x0`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

enable_lock_step

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

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x0`.

gicv3.BPR-min

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

Type: `int`. Default value: `0x2`.

gicv3.VBPR-min

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

Type: `int`. Default value: `0x2`.

has_dense_mem_map

If true, the cluster follows the dense memory map else it implements the sparse memory map.

Type: `bool`. Default value: `0x0`.

has_llpp

Equivalent to CFGLLPPIMP.

Type: `bool`. Default value: `0x1`.

has_spp

Equivalent to CFGSPPIMP.

Type: `bool`. Default value: `0x1`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

`icache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`icache-size`

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

`icache-state_modelled`

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

`l2cache-hit_latency`

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`l2cache-maintenance_latency`

L2 Cache timing annotation latency for cache maintenance operations given in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`l2cache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

`l2cache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`l2cache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`l2cache-size`

L2 Cache size in bytes.

Type: `int`. Default value: `0x400000`.

l2cache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

l2cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

l2cache-write_latency

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.

Type: `int`. Default value: `0x0`.

memory.ext_slave_base

Equivalent to CFGACELSTCMBASEADDR.

Type: `int`. Default value: `0x0`.

memory.has_llram

Equivalent to CFGLLRAMIMP.

Type: `bool`. Default value: `0x1`.

memory.llram_base

Equivalent to CFGLLRAMBASEADDR.

Type: `int`. Default value: `0x20000000`.

memory.llram_enable_at_reset

Equivalent to CFGLLRAMEN.

Type: `bool`. Default value: `0x1`.

memory.llram_shared

Equivalent to CFGLLRAMSHARED and it is only functional for revision 2.

Type: `bool`. Default value: `0x0`.

memory.llram_size

Size of the LLRAM.

Type: `int`. Default value: `0x10000000`.

num_protection_regions_s1

Number of v8-R protection regions.

Type: `int`. Default value: `0x10`.

num_protection_regions_s2

Number of v8-R hyp protection regions.

Type: `int`. Default value: `0x10`.

patch_level

Cosmetic change to Revision field in MIDR/MIDR_EL1. Corresponds to the patch number Y in rXpY.

Type: `int`. Default value: `0x1`.

PERIPHBASE

Base address of peripheral memory space.

Type: `int`. Default value: `0x13080000`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

ram_protection_enable_at_reset

Equivalent to CFGGRAMPROTEN.

Type: `bool`. Default value: `0x0`.

revision_number

Cosmetic change to Variant field in MIDR/MIDR_EL1. Corresponds to the revision number X in rXpY.

Type: `int`. Default value: `0x1`.

spp.base

Equivalent to CFGSPPBASEADDR.

Type: `int`. Default value: `0x0`.

spp.size

Sets the size of SPP(in bytes).

Type: `int`. Default value: `0x8000000`.

stage12_tlb_size

If VMSA is supported at stage1, number of stage1+2 tlb entries. If `instruction_tlb_size != 0`, this is treated as dtlb size.

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

VMSA_supported

VMSA is supported at EL1.

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.61 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 4-200: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM CortexX1CCT contains the following CADI targets:

- ARM_Cortex-X1C
- Cluster_ARM_Cortex-X1C
- PVMCache
- TlbCadi

ARM CortexX1CCT contains the following MTI components:

- [ARM_Cortex-X1C](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVMCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

About ARM CortexX1CCT

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 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 ARMCortexX1CCT

Table 4-201: 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.

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.
cpuporeset[8]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[8]	Signal	Slave	Disable cryptography extensions after reset.
cti[8]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[8]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[8]	Signal	Master	No power-down request.
dbgprupreq[8]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[8]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[8]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[8]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.

Name	Protocol	Type	Description
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 the location of the exception vectors at reset.
virq[8]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[8]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-X1C

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CFGTE

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

Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-write_access_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-write_latency

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.

Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).

Type: int. Default value: 0x100.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.
Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: `bool`. Default value: `0x0`.

cpu0.VINITHI

Reset value of SCTLR.V.
Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-X1C**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.
Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.
Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

dcache-snoop_data_transfer_latency

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.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_coherent_icache

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`. Default value: `0x0`.

has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x100000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

treat-dcache-cmos-to-pou-as-nop

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.

Type: `int`. Default value: `0x0`.

treat_PAC_as_NOP

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.

Type: `bool`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.62 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 4-202: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexX1CT contains the following CADI targets:

- ARM_Cortex-X1
- Cluster_ARM_Cortex-X1
- PVCache
- TlbCadi

ARMCortexX1CT contains the following MTI components:

- [ARM_Cortex-X1](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

About ARMCortexX1CT

The model supports the following features:

- DynamIQ™ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, or `nPMBIRQ` signals.
- DynamIQ™ features that are irrelevant to the programmers' view simulation, for example:
 - Automatic CPU retention mode.
 - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. The following models also exist, with a fixed number of cores per cluster:

- ARMCortexX1x1CT.
- ARMCortexX1x2CT.
- ARMCortexX1x3CT.
- ARMCortexX1x4CT.

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 ARMCortexX1CT

Table 4-203: 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.

Name	Protocol	Type	Description
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The 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.

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[4]	PChannel	Slave	PChannels for cores
pmbirq[4]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Cortex-X1

cpu0.CFGEND

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

Copyright © 2017–2023 Arm Limited (or its affiliates). All rights reserved.

Non-Confidential

Type: `bool`. Default value: `0x0`.

cpu0.CFGTE

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

Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-write_access_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-write_latency

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.

Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).

Type: int. Default value: 0x100.

cpu0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.
Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.
Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.
Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.
Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.
Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.
Type: `bool`. Default value: `0x0`.

cpu0.VINITHI

Reset value of SCTLR.V.
Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-X1**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.
Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.
Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_coherent_icache

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`. Default value: `0x0`.

has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x100000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

treat-dcache-cmos-to-pou-as-nop

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.

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.63 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 4-204: IP revisions support

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARM CortexX2CT contains the following CADI targets:

- ARM_Cortex-X2
- Cluster_ARM_Cortex-X2
- PVCache
- TlbCadi

ARMCortexX2CT contains the following MTI components:

- [ARM_Cortex-X2](#)
- [ARMv8Cluster](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PPUv1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

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 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

This model does not support the Power Control State Machine (PCSM).

Ports for ARMCortexX2CT

Table 4-205: 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_powerdown_out	Signal	Master	An output from PPUs that informs thermal controller of the core 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.

Name	Protocol	Type	Description
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_powerdown_out[12]	Signal	Master	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.
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.

Name	Protocol	Type	Description
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 ARM_Cortex-X2

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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.
Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).
Type: int. Default value: 0x100.

cpu0.min_sync_level

Force minimum syncLevel (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll).
Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.
Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.
Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.
Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.
Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Cortex-X2

BROADCASTATOMIC

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-snoop_data_transfer_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.
Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.
Type: `int`. Default value: `0x0`.

default_opmode

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.
Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.
Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).
Type: `bool`. Default value: `0x1`.

ete.CLAIMTAGS

Number of claim tags.
Type: `int`. Default value: `0x20`.

ete.MAX_INST_PER_Q

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

Type: `int`. Default value: `0x1`.

`ete.NumberOfRSPairs`

Number of resource selector pairs.

Type: `int`. Default value: `0x8`.

`ete.PIDR_CM0D`

TRCPIDR CM0D value.

Type: `int`. Default value: `0x0`.

`ete.PIDR_REVAND`

TRCPIDR REVAND value.

Type: `int`. Default value: `0x0`.

`ete.PIDR_REVISION`

TRCPIDR REVISION value.

Type: `int`. Default value: `0x0`.

`ete.Q_CADENCE`

Number of instruction blocks traced between two Q elements.

Type: `int`. Default value: `0x1`.

`ete.RES0_STATEFUL`

Whether RES0 bits are stateful or RAZ/WI.

Type: `bool`. Default value: `0x0`.

`ete.RETSTACK`

Return stack depth.

Type: `int`. Default value: `0x3`.

`ete.REVISION`

TRCIDR1 revision value.

Type: `int`. Default value: `0x2`.

`ete.SIM_OVERFLOW_GRANULARITY`

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

Type: `int`. Default value: `0x64`.

`ete.SIM_OVERFLOW_PERCENTAGE`

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

Type: `int`. Default value: `0x0`.

`ete.SOURCE_ADDRESS`

Allow generation of source address elements.

Type: `bool`. Default value: `0x0`.

`ete.TRACE_OUTPUT`

File to which to write trace byte stream.

Type: `string`. Default value: `""`.

`ete.TRCRSRTA_FORCED_EXCEP`

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`. Default value: `0x0`.

ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`. Default value: `0x0`.

force_mte_tag_access_razwi_and_ignore_tag_checks

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.

Type: `bool`. Default value: `0x0`.

force_zero_mpam_partid_and_pmg

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.

Type: `int`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_ete

If true, implements the Embedded Trace Extension (FEAT_ETE). This option is discarded if ete plugin is explicitly loaded (`--plugin` or `-P`).

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state_modelled=true.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state_modelled=true.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

icache-read_latency

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.

Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x0`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

memory_tagging_support_level

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).

Type: `int`. Default value: `0x2`.

NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

num_nodes

Number of transport nodes. Zero implies direct-connect configuration.

Type: `int`. Default value: `0x1`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

treat-dcache-cmos-to-pou-as-nop

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.

Type: `int`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.64 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 4-206: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexX3CT contains the following CADI targets:

- ARM_Cortex-X3
- Cluster_ARM_Cortex-X3
- PVCache
- TlbCadi

ARMCortexX3CT contains the following MTI components:

- [ARM_Cortex-X3](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PPUv1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters removed:

- `advsimd_bf16_support_level`
- `fault_unalign_to_unsupported_access`
- `has_16bit_vmids`

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 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.
- This model does not support the Power Control State Machine (PCSM).

Ports for ARMCortexX3CT

Table 4-207: 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_powerdown_out	Signal	Master	An output from PPU that informs thermal controller of the core 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.

Name	Protocol	Type	Description
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_powerdown_out[12]	Signal	Master	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.
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.

Name	Protocol	Type	Description
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 ARM_Cortex-X3

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

`cpu0.enable_trace_special_hlt_imm16`

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

`cpu0.force-fpsid`

Override the FPSID value.

Type: `bool`. Default value: `0x1`.

`cpu0.l2cache-hit_latency`

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-maintenance_latency`

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-miss_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-read_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-size`

L2 Cache size in bytes.

Type: `int`. Default value: `0x100000`.

`cpu0.l2cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-snoop_issue_latency`

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.RVBARADDR`

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

`cpu0.vfp-present`

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for Cluster_ARM_Cortex-X3

`AEND0_DEFAULT`

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `int`. Default value: `0x0`.

`AEND1_DEFAULT`

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `int`. Default value: `0x0`.

`AEND2_DEFAULT`

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `int`. Default value: `0x0`.

AEND3_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `int`. Default value: `0x0`.

ASTART0_DEFAULT

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `int`. Default value: `0x0`.

ASTART1_DEFAULT

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `int`. Default value: `0x0`.

ASTART2_DEFAULT

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `int`. Default value: `0x0`.

ASTART3_DEFAULT

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `int`. Default value: `0x0`.

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x0`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x0`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

CPUCFR

Value of CPU Configuration Register.

Type: `int`. Default value: `0x0`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.
Type: `int`. Default value: `0x8000`.

dcache-snoop_data_transfer_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.
Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.
Type: `int`. Default value: `0x0`.

default_opmode

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.
Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.
Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

error_record_feature_register

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,"INJ":0x0,"FRX":0x0,"UC":0x0,"UEU":0x0,"UER":0x0,"UEO":0x0,"DE":0x0,"CI":0x0,"TS":0x0,"Visibility":0x0}]
```

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".

Type: `string`. Default value:

```
[{"ED":0x2,"IMPDEF_3_2":0x0,"UI":0x2,"FI":0x2,"UE":0x1,"CFI":0x2,"CEC":0x2,"RP":0x1,"DUI":0x0,"Visibility":"Cluster"},
```

```
{ "ED":0x2,"IMPDEF_3_2":0x0,"UI":0x2,"FI":0x2,"UE":0x1,"CFI":0x2,"CEC":0x2,"RP":0x1,"DUI":0x0,"Visibility":"Cluster"}]
```

ete.CLAIMTAGS

Number of claim tags.

Type: `int`. Default value: `0x20`.

ete.MAX_INST_PER_Q

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

Type: `int`. Default value: `0x1`.

ete.NumberOfRSPairs

Number of resource selector pairs.

Type: `int`. Default value: `0x8`.

ete.PIDR_CMOD

TRCPIDR CMOD value.

Type: `int`. Default value: `0x0`.

ete.PIDR_REVAND

TRCPIDR REVAND value.

Type: `int`. Default value: `0x0`.

ete.PIDR_REVISION

TRCPIDR REVISION value.

Type: `int`. Default value: `0x0`.

ete.Q_CADENCE

Number of instruction blocks traced between two Q elements.

Type: `int`. Default value: `0x1`.

ete.RES0_STATEFUL

Whether RES0 bits are stateful or RAZ/WI.

Type: `bool`. Default value: `0x0`.

ete.RETSTACK

Return stack depth.

Type: `int`. Default value: `0x3`.

ete.REVISION

TRCIDR1 revision value.

Type: `int`. Default value: `0x0`.

ete.SIM_OVERFLOW_GRANULARITY

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

Type: `int`. Default value: `0x64`.

ete.SIM_OVERFLOW_PERCENTAGE

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

Type: `int`. Default value: `0x0`.

ete.SOURCE_ADDRESS

Allow generation of source address elements.

Type: `bool`. Default value: `0x0`.

ete.TRACE_OUTPUT

File to which to write trace byte stream.

Type: `string`. Default value: `""`.

ete.TRCRSRTA_FORCED_EXCEP

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`. Default value: `0x0`.

ext_abort_normal_noncacheable_read_is_sync

Synchronous reporting of normal noncacheable-read external aborts.

Type: `bool`. Default value: `0x1`.

ext_abort_so_write_ras_type

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.

Type: `int`. Default value: `0x2`.

force_mte_tag_access_razwi_and_ignore_tag_checks

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.

Type: `bool`. Default value: `0x0`.

force_zero_mpam_partid_and_pmg

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.

Type: `int`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_enhanced_pan

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.

Type: `int`. Default value: `0x2`.

has_ete

If true, implements the Embedded Trace Extension (FEAT_ETE). This option is discarded if ete plugin is explicitly loaded (`--plugin` or `-P`).

Type: `bool`. Default value: `0x0`.

has_large_va

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.

Type: `int`. Default value: `0x0`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.
Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.
Type: `int`. Default value: `0x8000`.

icache-state_modelled

Set whether I-cache has stateful implementation.
Type: `bool`. Default value: `0x0`.

instruction_tlb_size

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).
Type: `int`. Default value: `0x0`.

l3cache-has_mpam

L3 Cache has MPAM support.
Type: `bool`. Default value: `0x1`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x40000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-ways

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

Type: `int`. Default value: `0x10`.

l3cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

memory_tagging_support_level

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).

Type: `int`. Default value: `0x2`.

num_nodes

Number of transport nodes. Zero implies direct-connect configuration.

Type: `int`. Default value: `0x1`.

pmu-num_counters

Number of PMU counters implemented.

Type: `int`. Default value: `0x6`.

pseudo_fault_generation_feature_register

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,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.

Type: string. Default value:

```
"[{"OF":true,"UC":true,"UEU":false,"UER":false,"UEO":false,"DE":0x1,"CE":0x1,"CI":true,"ER":
{"OF":false,"UC":true,"UEU":false,"UER":false,"UEO":false,"DE":0x1,"CE":0x1,"CI":false,"ER":
```

ptw_latency

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

Type: int. Default value: 0x0.

stage12_tlb_size

Number of stage1+2 tlb entries.

Type: int. Default value: 0x80.

tlb_latency

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

Type: int. Default value: 0x0.

trace_physical_registers_when_host_virtualisation_enabled

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.

Type: int. Default value: 0x1.

treat_PAC_as_NOP

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.

Type: bool. Default value: 0x0.

walk_cache_latency

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

Type: int. Default value: 0x0.

4.5.65 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 4-208: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMCortexX4CT contains the following CADI targets:

- ARM_Cortex-X4
- Cluster_ARM_Cortex-X4
- PVCache
- TlbCadi

ARMCortexX4CT contains the following MTI components:

- [ARM_Cortex-X4](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PPUv1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `cpu0.l2cache-hit_latency`
- `cpu0.l2cache-maintenance_latency`
- `cpu0.l2cache-miss_latency`
- `cpu0.l2cache-read_access_latency`
- `cpu0.l2cache-read_latency`
- `cpu0.l2cache-snoop_data_transfer_latency`
- `cpu0.l2cache-snoop_issue_latency`
- `cpu0.l2cache-write_access_latency`
- `cpu0.l2cache-write_latency`
- `cpu1.l2cache-hit_latency`

- `cpu1.l2cache-maintenance_latency`
- `cpu1.l2cache-miss_latency`
- `cpu1.l2cache-read_access_latency`
- `cpu1.l2cache-read_latency`
- `cpu1.l2cache-snoop_data_transfer_latency`
- `cpu1.l2cache-snoop_issue_latency`
- `cpu1.l2cache-write_access_latency`
- `cpu1.l2cache-write_latency`
- `cpu10.l2cache-hit_latency`
- `cpu10.l2cache-maintenance_latency`
- `cpu10.l2cache-miss_latency`
- `cpu10.l2cache-read_access_latency`
- `cpu10.l2cache-read_latency`
- `cpu10.l2cache-snoop_data_transfer_latency`
- `cpu10.l2cache-snoop_issue_latency`
- `cpu10.l2cache-write_access_latency`
- `cpu10.l2cache-write_latency`
- `cpu11.l2cache-hit_latency`
- `cpu11.l2cache-maintenance_latency`
- `cpu11.l2cache-miss_latency`
- `cpu11.l2cache-read_access_latency`
- `cpu11.l2cache-read_latency`
- `cpu11.l2cache-snoop_data_transfer_latency`
- `cpu11.l2cache-snoop_issue_latency`
- `cpu11.l2cache-write_access_latency`
- `cpu11.l2cache-write_latency`
- `cpu12.l2cache-hit_latency`
- `cpu12.l2cache-maintenance_latency`
- `cpu12.l2cache-miss_latency`
- `cpu12.l2cache-read_access_latency`
- `cpu12.l2cache-read_latency`
- `cpu12.l2cache-snoop_data_transfer_latency`
- `cpu12.l2cache-snoop_issue_latency`
- `cpu12.l2cache-write_access_latency`

- `cpu12.l2cache-write_latency`
- `cpu13.l2cache-hit_latency`
- `cpu13.l2cache-maintenance_latency`
- `cpu13.l2cache-miss_latency`
- `cpu13.l2cache-read_access_latency`
- `cpu13.l2cache-read_latency`
- `cpu13.l2cache-snoop_data_transfer_latency`
- `cpu13.l2cache-snoop_issue_latency`
- `cpu13.l2cache-write_access_latency`
- `cpu13.l2cache-write_latency`
- `cpu2.l2cache-hit_latency`
- `cpu2.l2cache-maintenance_latency`
- `cpu2.l2cache-miss_latency`
- `cpu2.l2cache-read_access_latency`
- `cpu2.l2cache-read_latency`
- `cpu2.l2cache-snoop_data_transfer_latency`
- `cpu2.l2cache-snoop_issue_latency`
- `cpu2.l2cache-write_access_latency`
- `cpu2.l2cache-write_latency`
- `cpu3.l2cache-hit_latency`
- `cpu3.l2cache-maintenance_latency`
- `cpu3.l2cache-miss_latency`
- `cpu3.l2cache-read_access_latency`
- `cpu3.l2cache-read_latency`
- `cpu3.l2cache-snoop_data_transfer_latency`
- `cpu3.l2cache-snoop_issue_latency`
- `cpu3.l2cache-write_access_latency`
- `cpu3.l2cache-write_latency`
- `cpu4.l2cache-hit_latency`
- `cpu4.l2cache-maintenance_latency`
- `cpu4.l2cache-miss_latency`
- `cpu4.l2cache-read_access_latency`
- `cpu4.l2cache-read_latency`
- `cpu4.l2cache-snoop_data_transfer_latency`

- `cpu4.l2cache-snoop_issue_latency`
- `cpu4.l2cache-write_access_latency`
- `cpu4.l2cache-write_latency`
- `cpu5.l2cache-hit_latency`
- `cpu5.l2cache-maintenance_latency`
- `cpu5.l2cache-miss_latency`
- `cpu5.l2cache-read_access_latency`
- `cpu5.l2cache-read_latency`
- `cpu5.l2cache-snoop_data_transfer_latency`
- `cpu5.l2cache-snoop_issue_latency`
- `cpu5.l2cache-write_access_latency`
- `cpu5.l2cache-write_latency`
- `cpu6.l2cache-hit_latency`
- `cpu6.l2cache-maintenance_latency`
- `cpu6.l2cache-miss_latency`
- `cpu6.l2cache-read_access_latency`
- `cpu6.l2cache-read_latency`
- `cpu6.l2cache-snoop_data_transfer_latency`
- `cpu6.l2cache-snoop_issue_latency`
- `cpu6.l2cache-write_access_latency`
- `cpu6.l2cache-write_latency`
- `cpu7.l2cache-hit_latency`
- `cpu7.l2cache-maintenance_latency`
- `cpu7.l2cache-miss_latency`
- `cpu7.l2cache-read_access_latency`
- `cpu7.l2cache-read_latency`
- `cpu7.l2cache-snoop_data_transfer_latency`
- `cpu7.l2cache-snoop_issue_latency`
- `cpu7.l2cache-write_access_latency`
- `cpu7.l2cache-write_latency`
- `cpu8.l2cache-hit_latency`
- `cpu8.l2cache-maintenance_latency`
- `cpu8.l2cache-miss_latency`
- `cpu8.l2cache-read_access_latency`

- `cpu8.l2cache-read_latency`
- `cpu8.l2cache-snoop_data_transfer_latency`
- `cpu8.l2cache-snoop_issue_latency`
- `cpu8.l2cache-write_access_latency`
- `cpu8.l2cache-write_latency`
- `cpu9.l2cache-hit_latency`
- `cpu9.l2cache-maintenance_latency`
- `cpu9.l2cache-miss_latency`
- `cpu9.l2cache-read_access_latency`
- `cpu9.l2cache-read_latency`
- `cpu9.l2cache-snoop_data_transfer_latency`
- `cpu9.l2cache-snoop_issue_latency`
- `cpu9.l2cache-write_access_latency`
- `cpu9.l2cache-write_latency`

Parameters removed:

- `advsimd_bf16_support_level`

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.
- This model does not support the Power Control State Machine (PCSM).

Ports for ARMCortexX4CT

Table 4-209: 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).
<code>AEND2MP</code>	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_powerdown_out	Signal	Master	An output from PPU that informs thermal controller of the core 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_powerdown_out[14]	Signal	Master	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.
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.

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 ARM_Cortex-X4

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CFGTE

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-miss_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-ways

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

Type: `int`. Default value: `0x8`.

cpu0.l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.min_sync_level

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

Type: `int`. Default value: `0x0`.

cpu0.RVBARADDR

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

cpu0.semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

cpu0.semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

cpu0.semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

cpu0.semihosting-stack_limit

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

cpu0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

cpu0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

cpu0.trace_special_hlt_imm16

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for Cluster_ARM_Cortex-X4

AEND0_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `int`. Default value: `0x0`.

AEND1_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `int`. Default value: `0x0`.

AEND2_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `int`. Default value: `0x0`.

AEND3_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `int`. Default value: `0x0`.

ASTART0_DEFAULT

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `int`. Default value: `0x0`.

ASTART1_DEFAULT

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `int`. Default value: `0x0`.

ASTART2_DEFAULT

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `int`. Default value: `0x0`.

ASTART3_DEFAULT

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `int`. Default value: `0x0`.

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x0`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x0`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

CPUCFR

Value of CPU Configuration Register.

Type: `int`. Default value: `0x0`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

ecv_support_level

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT_ECV).

Type: `int`. Default value: `0x2`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

ete.CLAIMTAGS

Number of claim tags.

Type: `int`. Default value: `0x4`.

ete.MAX_INST_PER_Q

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

Type: `int`. Default value: `0x1`.

ete.NumberOfRSPairs

Number of resource selector pairs.

Type: `int`. Default value: `0x8`.

ete.PIDR_CM0D

TRCPIDR CM0D value.

Type: `int`. Default value: `0x0`.

ete.Q_CADENCE

Number of instruction blocks traced between two Q elements.

Type: `int`. Default value: `0x1`.

ete.RES0_STATEFUL

Whether RES0 bits are stateful or RAZ/WI.

Type: `bool`. Default value: `0x0`.

ete.RETSTACK

Return stack depth.

Type: `int`. Default value: `0x3`.

ete.SIM_OVERFLOW_GRANULARITY

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

Type: `int`. Default value: `0x64`.

ete.SIM_OVERFLOW_PERCENTAGE

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

Type: `int`. Default value: `0x0`.

ete.SOURCE_ADDRESS

Allow generation of source address elements.

Type: `bool`. Default value: `0x0`.

ete.TRACE_OUTPUT

File to which to write trace byte stream.

Type: `string`. Default value: `""`.

ete.TRCRSRTA_FORCED_EXCEP

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`. Default value: `0x0`.

ext_abort_so_write_ras_type

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.

Type: `int`. Default value: `0x2`.

force_mte_tag_access_razwi_and_ignore_tag_checks

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.

Type: `bool`. Default value: `0x0`.

force_zero_mpam_partid_and_pmg

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.

Type: `int`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_enhanced_pan

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.

Type: `int`. Default value: `0x2`.

has_ete

If true, implements the Embedded Trace Extension (FEAT_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

has_statistical_profiling

Whether Statistical Based Profiling is implemented (FEAT_SPE).

Type: `bool`. Default value: `0x1`.

has_v8_7_spe_inverted_filtering

Where FEAT_SPEv1p2 is implemented, whether the inverting filtering by events.

Type: `bool`. Default value: `0x0`.

has_v8_7_spe_previous_branch_target

Where FEAT_SPEv1p2 is implemented, whether the optional branch target feature is implemented.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

instruction_tlb_size

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

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

l3cache-ways

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

Type: `int`. Default value: `0x10`.

log2_trace_buffer_alignment

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

Type: `int`. Default value: `0x6`.

memory_tagging_support_level

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).

Type: `int`. Default value: `0x3`.

mpam_has_altsp

MPAM Whether MPAMIDR_EL1.HAS_ALTSP bit is set or clear.

Type: `bool`. Default value: `0x0`.

mpamidr_has_force_ns

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.

Type: `int`. Default value: `0x0`.

mpamidr_has_sdeflt

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.

Type: `int`. Default value: `0x1`.

mpamidr_has_tidr

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.

Type: `int`. Default value: `0x1`.

num_acp

Number of ACP ports.

Type: `int`. Default value: `0x0`.

NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

num_nodes

Number of transport nodes. Zero implies direct-connect configuration.

Type: `int`. Default value: `0x1`.

pmu-num_counters

Number of PMU counters implemented.

Type: `int`. Default value: `0x1f`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

stage12_tlb_size

Number of stage1+2 tlb entries.

Type: `int`. Default value: `0x80`.

tcr_txsz_undersize_should_fault

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

treat_PAC_as_NOP

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.

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.66 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 4-210: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMNeoverseE1CT contains the following CADI targets:

- `ARM_Neoverse-E1`
- `Cluster_ARM_Neoverse-E1`
- `PVCache`
- `TlbCadi`

ARMNeoverseE1CT contains the following MTI components:

- [ARM_Neoverse-E1](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)

- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

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 following models also exist, with a fixed number of cores per cluster:

- ARMNeoverseE1x1CT.
- ARMNeoverseE1x2CT.
- ARMNeoverseE1x3CT.
- ARMNeoverseE1x4CT.

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 ARMNeoverseE1CT

Table 4-211: Ports

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[16]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[16]	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.
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.

Name	Protocol	Type	Description
CNTPSIRQ[16]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[16]	Signal	Master	Timer signals to SOC.
commirq[16]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[8]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable[16]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[8]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[16]	Signal	Slave	Disable cryptography extensions after reset.
cti[16]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[16]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[16]	Signal	Master	These signals relate to core power down.
dbgpwrupreq[16]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[16]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
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.

Name	Protocol	Type	Description
rvbaraddr[16]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[16]	Signal	Slave	Per core virtual System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[16]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[16]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[16]	Signal	Slave	Virtualised FIQ.
vinithi[16]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[16]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[16]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Neoverse-E1

cpu0.thread0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.thread0.enable_single_thread_at_reset

Enable single thread after reset and keep other thread in reset.
Type: bool. Default value: 0x0.

cpu0.thread0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.
Type: bool. Default value: 0x0.

cpu0.thread0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.
Type: int. Default value: 0x0.

cpu0.thread0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.
Type: int. Default value: 0x0.

cpu0.thread0.l2cache-miss_latency

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.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x40000`.

cpu0.thread0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks.

This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.l2cache-write_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.thread0.max_code_cache_mb

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).

Type: `int`. Default value: `0x100`.

cpu0.thread0.min_sync_level

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

Type: int. Default value: 0x0.

cpu0.thread0.semihosting-A32_HLT

A32 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.thread0.semihosting-A64_HLT

A64 HLT number for semihosting calls.

Type: int. Default value: 0xf000.

cpu0.thread0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.

Type: int. Default value: 0x123456.

cpu0.thread0.semihosting-cmd_line

Command line available to semihosting calls.

Type: string. Default value: "".

cpu0.thread0.semihosting-cwd

Base directory for semihosting file access.

Type: string. Default value: "".

cpu0.thread0.semihosting-enable

Enable semihosting SVC/HLT traps.

Type: bool. Default value: 0x1.

cpu0.thread0.semihosting-heap_base

Virtual address of heap base.

Type: int. Default value: 0x0.

cpu0.thread0.semihosting-heap_limit

Virtual address of top of heap.

Type: int. Default value: 0xf000000.

cpu0.thread0.semihosting-stack_base

Virtual address of base of descending stack.

Type: int. Default value: 0x10000000.

cpu0.thread0.semihosting-stack_limit

Virtual address of stack limit.

Type: int. Default value: 0xf000000.

cpu0.thread0.semihosting-T32_HLT

T32 HLT number for semihosting calls.

Type: int. Default value: 0x3c.

cpu0.thread0.semihosting-Thumb_SVC

T32 SVC number for semihosting calls.

Type: int. Default value: 0xab.

cpu0.thread0.trace_special_hlt_imm16

For this HLT number, IF enable_trace_special_hlt_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

cpu0.thread0.vfp-enable_at_reset

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

cpu0.thread0.vfp-present

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for Cluster_ARM_Neoverse-E1**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

cluster_patch_level

Cosmetic change to patch number in CLUSTERIDR. Corresponds to the Y in rXpY.

Type: `int`. Default value: `0x0`.

cluster_revision_number

Cosmetic change to revision number in CLUSTERIDR, Corresponds to the X in rXpY.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

force_zero_PSTATE_PAN

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`.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_dot_product

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

Type: `int`. Default value: `0x2`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.

Type: `int`. Default value: `0x400000`.

l3cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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.

Type: `int`. Default value: `0x0`.

l3cache-write_latency

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.

Type: `int`. Default value: `0x0`.

NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.67 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 4-212: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMNeoverseN1CT contains the following CADI targets:

- ARM_Neoverse-N1
- Cluster_ARM_Neoverse-N1
- PVCache
- TlbCadi

ARMNeoverseN1CT contains the following MTI components:

- [ARM_Neoverse-N1](#)
- [ARMv8Cluster](#)
- [AsyncCacheFlushUnit](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

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 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 following models also exist, with a fixed number of cores per cluster:

- ARMNeoverseN1x1CT.
- ARMNeoverseN1x2CT.
- ARMNeoverseN1x3CT.
- ARMNeoverseN1x4CT.

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 4-213: Ports

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.

Name	Protocol	Type	Description
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The 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.

Name	Protocol	Type	Description
pchannel_core[4]	PChannel	Slave	PChannels for cores
pmbirq[4]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

Parameters for ARM_Neoverse-N1

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CFGTE

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

Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x80000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.RVBARADDR`

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0xf000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf0000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

`cpu0.VINITHI`

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Neoverse-N1

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAIN

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.

Type: `int`. Default value: `0x0`.

default_opmode

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.

Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.

Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).

Type: `bool`. Default value: `0x1`.

ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_coherent_icache

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`. Default value: `0x0`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

has_statistical_profiling

Whether Statistical Based Profiling is implemented (FEAT_SPE).

Type: `bool`. Default value: `0x1`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

icache-read_latency

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.

Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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.

Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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.

Type: `int`. Default value: `0x0`.

`l3cache-read_latency`

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`.
Type: `int`. Default value: `0x0`.

`l3cache-size`

L3 Cache size in bytes.
Type: `int`. Default value: `0x100000`.

`l3cache-snoop_data_transfer_latency`

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`.
Type: `int`. Default value: `0x0`.

`l3cache-snoop_issue_latency`

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

`l3cache-write_access_latency`

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`.
Type: `int`. Default value: `0x0`.

`l3cache-write_latency`

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`.
Type: `int`. Default value: `0x0`.

`NUM_CORES`

Number of cores per cluster.
Type: `int`. Default value: `0x1`.

`pchannel_treat_simreset_as_poreset`

Register core as ON state to cluster with simulation reset.
Type: `bool`. Default value: `0x0`.

`periph_address_end`

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).
Type: `int`. Default value: `0x0`.

`periph_address_start`

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

Type: `int`. Default value: `0x0`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

treat-dcache-cmos-to-pou-as-nop

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.

Type: `int`. Default value: `0x0`.

use_architectural_names

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

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.68 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 4-214: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMNeoverseN2CT contains the following CADI targets:

- `ARM_Neoverse-N2`
- `Cluster_ARM_Neoverse-N2`
- `PVCache`
- `TlbCadi`

ARMNeoverseN2CT contains the following MTI components:

- [ARM_Neoverse-N2](#)
- [ARMv8Cluster](#)
- [DSU](#)

- [GICv3CPUInterfaceDecoder](#)
- [PPUv1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

About ARMNeoverseN2CT

The model supports the following features:

- DynamIQ™ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

Support for the following features is planned for a future release:

- DynamIQ™ Shared Unit-110 (DSU-110) system registers.
- `BROADCASTCACHEMAINTPOU` pin
- `COREINSTRRET`, `COREINSTRRUN`, and `nPMBIRQ` signals
- DSU-110 cluster. The implementation relies on DynamIQ™ only.
- `TRBE`.
- Transporting architectural metadata tags over TLM2 (OSCI) and AMBA-PV (the Arm TLM2 extensions) bus protocols.

The following features will not be implemented:

- 256-bit wide output transactions.
- Error correction/detection features.
- Self-test features (MBIST).
- Snoop filtering.
- Armv9 trace extensions.

This model supports the 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

This model does not support the Power Control State Machine (PCSM).

Ports for ARMNeoverseN2CT

Table 4-215: 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_powerdown_out	Signal	Master	An output from PPU that informs thermal controller of the core 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.

Name	Protocol	Type	Description
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_powerdown_out[1]	Signal	Master	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.
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.

Name	Protocol	Type	Description
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 ARM_Neoverse-N2

cpu0.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian.
Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.
Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.
Type: bool. Default value: 0x0.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-size

L2 Cache size in bytes.

Type: `int`. Default value: `0x1000000`.

cpu0.l2cache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-snoop_issue_latency

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-write_latency

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.
Type: int. Default value: 0x0.

cpu0.max_code_cache_mb

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).
Type: int. Default value: 0x100.

cpu0.min_sync_level

Force minimum syncLevel (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll).
Type: int. Default value: 0x0.

cpu0.RVBARADDR

Value of RVBAR_ELx register.
Type: int. Default value: 0x0.

cpu0.semihosting-A32_HLT

A32 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-A64_HLT

A64 HLT number for semihosting calls.
Type: int. Default value: 0xf000.

cpu0.semihosting-ARM_SVC

A32 SVC number for semihosting calls.
Type: int. Default value: 0x123456.

cpu0.semihosting-cmd_line

Command line available to semihosting calls.
Type: string. Default value: "".

cpu0.semihosting-cwd

Base directory for semihosting file access.
Type: string. Default value: "".

cpu0.semihosting-enable

Enable semihosting SVC/HLT traps.
Type: bool. Default value: 0x1.

cpu0.semihosting-heap_base

Virtual address of heap base.
Type: int. Default value: 0x0.

cpu0.semihosting-heap_limit

Virtual address of top of heap.
Type: int. Default value: 0xf000000.

cpu0.semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Neoverse-N2

BROADCASTATOMIC

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x1`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x1`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`. Default value: `0x1`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-snoop_data_transfer_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.
Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.
Type: `int`. Default value: `0x0`.

dcache-write_latency

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`.
Type: `int`. Default value: `0x0`.

default_opmode

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.
Type: `int`. Default value: `0x4`.

diagnostics

Enable DynamIQ diagnostic messages.
Type: `bool`. Default value: `0x0`.

enable_simulation_performance_optimizations

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).
Type: `bool`. Default value: `0x1`.

ete.CLAIMTAGS

Number of claim tags.
Type: `int`. Default value: `0x20`.

ete.MAX_INST_PER_Q

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

Type: `int`. Default value: `0x1`.

`ete.NumberOfRSPairs`

Number of resource selector pairs.

Type: `int`. Default value: `0x8`.

`ete.PIDR_CM0D`

TRCPIDR CM0D value.

Type: `int`. Default value: `0x0`.

`ete.PIDR_REVAND`

TRCPIDR REVAND value.

Type: `int`. Default value: `0x0`.

`ete.PIDR_REVISION`

TRCPIDR REVISION value.

Type: `int`. Default value: `0x0`.

`ete.Q_CADENCE`

Number of instruction blocks traced between two Q elements.

Type: `int`. Default value: `0x1`.

`ete.RES0_STATEFUL`

Whether RES0 bits are stateful or RAZ/WI.

Type: `bool`. Default value: `0x0`.

`ete.RETSTACK`

Return stack depth.

Type: `int`. Default value: `0x3`.

`ete.REVISION`

TRCIDR1 revision value.

Type: `int`. Default value: `0x0`.

`ete.SIM_OVERFLOW_GRANULARITY`

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

Type: `int`. Default value: `0x64`.

`ete.SIM_OVERFLOW_PERCENTAGE`

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

Type: `int`. Default value: `0x0`.

`ete.SOURCE_ADDRESS`

Allow generation of source address elements.

Type: `bool`. Default value: `0x0`.

`ete.TRACE_OUTPUT`

File to which to write trace byte stream.

Type: `string`. Default value: `""`.

`ete.TRCRSRTA_FORCED_EXCEP`

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`. Default value: `0x0`.

ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`. Default value: `0x0`.

ext_abort_so_read_ras_type

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.

Type: `int`. Default value: `0x0`.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`. Default value: `0x0`.

force_mte_tag_access_razwi_and_ignore_tag_checks

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.

Type: `bool`. Default value: `0x0`.

force_zero_mpam_partid_and_pmg

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.

Type: `int`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_coherent_icache

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`. Default value: `0x0`.

has_ete

If true, implements the Embedded Trace Extension (FEAT_ETE). This option is discarded if ete plugin is explicitly loaded (`--plugin` or `-P`).

Type: `bool`. Default value: `0x0`.

has_external_rndr

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.

Type: `int`. Default value: `0x1`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

has_rndr

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT_RNG).

Type: `int`. Default value: `0x1`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.
Type: `int`. Default value: `0x0`.

icache-state_modelled

Set whether I-cache has stateful implementation.
Type: `bool`. Default value: `0x0`.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

l3cache-miss_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-read_access_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-read_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-snoop_data_transfer_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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.

Type: int. Default value: 0x0.

l3cache-write_latency

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.

Type: int. Default value: 0x0.

memory_tagging_support_level

Equivalent to BROADCASTMTE.

Type: int. Default value: 0x2.

mpam_max_partid

MPAM Maximum PARTID Supported.

Type: int. Default value: 0x1ff.

NUM_CORES

Number of cores per cluster.

Type: int. Default value: 0x1.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.

Type: bool. Default value: 0x0.

periph_address_end

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: int. Default value: 0x0.

periph_address_start

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: int. Default value: 0x0.

ptw_latency

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

Type: int. Default value: 0x0.

rndr_rndrrs_seed

Initial seed for random engine used in RNDR register.

Type: int. Default value: 0x0.

tlb_latency

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

Type: int. Default value: 0x0.

treat-dcache-cmos-to-pou-as-nop

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.

Type: `int`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.69 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 4-216: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMNeoverseV1CT contains the following CADI targets:

- ARM_Neoverse-V1
- Cluster_ARM_Neoverse-V1
- PVCache
- TlbCadi

ARMNeoverseV1CT contains the following MTI components:

- [ARM_Neoverse-V1](#)
- [ARMv8Cluster](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters added:

- `use_architectural_names`

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.

The following models also exist, with a fixed number of cores per cluster:

- ARMNeoverseV1x1CT.
- ARMNeoverseV1x2CT.
- ARMNeoverseV1x4CT.

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 4-217: 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_Neoverse-V1

cpu0.CFGEND

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

Type: `bool`. Default value: `0x0`.

cpu0.CFGTE

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

Type: `bool`. Default value: `0x0`.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`. Default value: `0x0`.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`. Default value: `0x0`.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

cpu0.l2cache-read_latency

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`.
Type: `int`. Default value: `0x0`.

`cpu0.l2cache-size`

L2 Cache size in bytes.
Type: `int`. Default value: `0x80000`.

`cpu0.l2cache-snoop_data_transfer_latency`

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`.
Type: `int`. Default value: `0x0`.

`cpu0.l2cache-snoop_issue_latency`

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_access_latency`

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`.
Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_latency`

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`.
Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).
Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

Force minimum syncLevel (0=off=default,1=syncState,2=postInsnIO,3=postInsnAll).
Type: `int`. Default value: `0x0`.

`cpu0.RVBARADDR`

Value of RVBAR_ELx register.
Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.
Type: `int`. Default value: `0x1000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.
Type: `int`. Default value: `0x1000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

`cpu0.VINITHI`

Reset value of SCTLR.V.

Type: `bool`. Default value: `0x0`.

Parameters for Cluster_ARM_Neoverse-V1

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: bool. Default value: 0x1.

BROADCASTCACHEMAINT

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

Type: bool. Default value: 0x1.

BROADCASTOUTER

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

Type: bool. Default value: 0x1.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: bool. Default value: 0x1.

CLUSTER_ID

Processor cluster ID value.

Type: int. Default value: 0x0.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: int. Default value: 0x1.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: int. Default value: 0x1.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: int. Default value: 0x1.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x10000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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.

Type: int. Default value: 0x0.

default_opmode

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.

Type: int. Default value: 0x4.

diagnostics

Enable DynamIQ diagnostic messages.

Type: bool. Default value: 0x0.

enable_simulation_performance_optimizations

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).

Type: bool. Default value: 0x1.

enhanced_pac2_level

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).

Type: int. Default value: 0x1.

ext_abort_device_read_is_sync

Synchronous reporting of device-nGnRE read external aborts.

Type: bool. Default value: 0x0.

ext_abort_device_write_is_sync

Synchronous reporting of device-nGnRE write external aborts.

Type: bool. Default value: 0x0.

ext_abort_so_read_is_sync

Synchronous reporting of device-nGnRnE read external aborts.

Type: bool. Default value: 0x0.

ext_abort_so_write_is_sync

Synchronous reporting of device-nGnRnE write external aborts.

Type: bool. Default value: 0x0.

force_zero_mpam_partid_and_pmg

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.

Type: `int`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_acp

If true, Accelerator Coherency Port is configured.

Type: `bool`. Default value: `0x0`.

has_coherent_icache

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`. Default value: `0x1`.

has_external_rndr

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.

Type: `int`. Default value: `0x1`.

has_peripheral_port

If true, additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

has_rndr

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT_RNG).

Type: `int`. Default value: `0x1`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: bool. Default value: 0x0.

icache-read_access_latency

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.

Type: int. Default value: 0x0.

icache-read_latency

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.

Type: int. Default value: 0x0.

icache-size

L1 I-Cache size in bytes.

Type: int. Default value: 0x10000.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: bool. Default value: 0x0.

l3cache-hit_latency

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

l3cache-maintenance_latency

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

l3cache-miss_latency

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.

Type: int. Default value: 0x0.

l3cache-read_access_latency

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.

Type: int. Default value: 0x0.

l3cache-read_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-size

L3 Cache size in bytes.
Type: `int`. Default value: `0x0`.

l3cache-snoop_data_transfer_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-snoop_issue_latency

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.
Type: `int`. Default value: `0x0`.

l3cache-write_access_latency

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`.
Type: `int`. Default value: `0x0`.

l3cache-write_latency

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`.
Type: `int`. Default value: `0x0`.

MIDR

Value of `MIDR_EL1` register.
Type: `int`. Default value: `0x410fd400`.

pchannel_treat_simreset_as_poreset

Register core as ON state to cluster with simulation reset.
Type: `bool`. Default value: `0x0`.

periph_address_end

End address for peripheral port address range exclusive (corresponds to `AENDMP` input signal).
Type: `int`. Default value: `0x0`.

periph_address_start

Start address for peripheral port address range inclusive (corresponds to `ASTARTMP` input signal).
Type: `int`. Default value: `0x0`.

ptw_latency

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

tlb_latency

TLB latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

treat-dcache-cmos-to-pou-as-nop

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.
Type: `int`. Default value: `0x0`.

use_architectural_names

Use names SP/LR/PC instead of R13/R14/R15.
Type: `bool`. Default value: `0x0`.

walk_cache_latency

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.
Type: `int`. Default value: `0x0`.

4.5.70 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 4-218: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMNeoverseV2CT contains the following CADI targets:

- ARM_Neoverse-V2
- Cluster_ARM_Neoverse-V2
- PVCache
- TlbCadi

ARMNeoverseV2CT contains the following MTI components:

- [ARM_Neoverse-V2](#)
- [ARMv8Cluster](#)
- [DSU](#)
- [GICv3CPUInterfaceDecoder](#)

- [PPUv1](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)
- [TLB](#)

Changes in 11.24.4

Parameters removed:

- `fault_unalign_to_unsupported_access`
- `has_16bit_vmids`

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 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](#).

The following models also exist, with a fixed number of cores per cluster:

- ARMNeoverseV2x1CT.

Limitations

This model does not support the Power Control State Machine (PCSM).

Ports for ARMNeoverseV2CT

Table 4-219: 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.

Name	Protocol	Type	Description
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_powerdown_out	Signal	Master	An output from PPU that informs thermal controller of the core 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_powerdown_out[1]	Signal	Master	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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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 ARM_Neoverse-V2

cpu0.CFGEND

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

Type: bool. Default value: 0x0.

cpu0.CRYPTODISABLE

Disable cryptographic features.

Type: bool. Default value: 0x0.

cpu0.enable_trace_special_hlt_imm16

Enable usage of parameter trace_special_hlt_imm16.

Type: bool. Default value: 0x0.

cpu0.force-fpsid

Override the FPSID value.

Type: bool. Default value: 0x1.

cpu0.l2cache-hit_latency

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-maintenance_latency

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state_modelled=true.

Type: int. Default value: 0x0.

cpu0.l2cache-miss_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_access_latency

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.

Type: int. Default value: 0x0.

cpu0.l2cache-read_latency

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-size`

L2 Cache size in bytes.

Type: `int`. Default value: `0x100000`.

`cpu0.l2cache-snoop_data_transfer_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-snoop_issue_latency`

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_access_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.l2cache-write_latency`

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`.

Type: `int`. Default value: `0x0`.

`cpu0.max_code_cache_mb`

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).

Type: `int`. Default value: `0x100`.

`cpu0.min_sync_level`

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

Type: `int`. Default value: `0x0`.

`cpu0.RVBARADDR`

Value of RVBAR_ELx register.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-A32_HLT`

A32 HLT number for semihosting calls.

Type: `int`. Default value: `0x1000`.

`cpu0.semihosting-A64_HLT`

A64 HLT number for semihosting calls.

Type: `int`. Default value: `0x1000`.

`cpu0.semihosting-ARM_SVC`

A32 SVC number for semihosting calls.

Type: `int`. Default value: `0x123456`.

`cpu0.semihosting-cmd_line`

Command line available to semihosting calls.

Type: `string`. Default value: `""`.

`cpu0.semihosting-cwd`

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

`cpu0.semihosting-enable`

Enable semihosting SVC/HLT traps.

Type: `bool`. Default value: `0x1`.

`cpu0.semihosting-heap_base`

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

`cpu0.semihosting-heap_limit`

Virtual address of top of heap.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-stack_base`

Virtual address of base of descending stack.

Type: `int`. Default value: `0x10000000`.

`cpu0.semihosting-stack_limit`

Virtual address of stack limit.

Type: `int`. Default value: `0xf000000`.

`cpu0.semihosting-T32_HLT`

T32 HLT number for semihosting calls.

Type: `int`. Default value: `0x3c`.

`cpu0.semihosting-Thumb_SVC`

T32 SVC number for semihosting calls.

Type: `int`. Default value: `0xab`.

`cpu0.trace_special_hlt_imm16`

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `int`. Default value: `0xf000`.

`cpu0.vfp-enable_at_reset`

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`. Default value: `0x0`.

`cpu0.vfp-present`

Set whether the model has VFP support.

Type: `bool`. Default value: `0x1`.

Parameters for Cluster_ARM_Neoverse-V2

AEND0_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `int`. Default value: `0x0`.

AEND1_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `int`. Default value: `0x0`.

AEND2_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `int`. Default value: `0x0`.

AEND3_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `int`. Default value: `0x0`.

ASTART0_DEFAULT

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `int`. Default value: `0x0`.

ASTART1_DEFAULT

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `int`. Default value: `0x0`.

ASTART2_DEFAULT

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `int`. Default value: `0x0`.

ASTART3_DEFAULT

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `int`. Default value: `0x0`.

BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`. Default value: `0x1`.

BROADCASTCACHEMAINT

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

Type: `bool`. Default value: `0x0`.

BROADCASTOUTER

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

Type: `bool`. Default value: `0x0`.

BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`. Default value: `0x1`.

bus_type

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `int`. Default value: `0x0`.

CLUSTER_ID

Processor cluster ID value.

Type: `int`. Default value: `0x0`.

CMO_broadcast_when_cache_state_modelling_disabled

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 .

Type: `int`. Default value: `0x1`.

core_power_on_by_default

If true, The cluster and core 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.

Type: `bool`. Default value: `0x0`.

cpi_div

Divider for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

cpi_mul

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `int`. Default value: `0x1`.

CPUCFR

Value of CPU Configuration Register.

Type: `int`. Default value: `0x0`.

dcache-hit_latency

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-maintenance_latency

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

dcache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

dcache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-read_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-size

L1 D-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

dcache-snoop_data_transfer_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-state_modelled

Set whether D-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

dcache-write_access_latency

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`.

Type: `int`. Default value: `0x0`.

dcache-write_latency

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.

Type: int. Default value: 0x0.

default_opmode

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.

Type: int. Default value: 0x4.

diagnostics

Enable DynamIQ diagnostic messages.

Type: bool. Default value: 0x0.

enable_simulation_performance_optimizations

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).

Type: bool. Default value: 0x1.

ete.CLAIMTAGS

Number of claim tags.

Type: int. Default value: 0x20.

ete.MAX_INST_PER_Q

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

Type: int. Default value: 0x1.

ete.NumberOfRSPairs

Number of resource selector pairs.

Type: int. Default value: 0x8.

ete.PIDR_CM0D

TRCPIDR CM0D value.

Type: int. Default value: 0x0.

ete.PIDR_REVAND

TRCPIDR REVAND value.

Type: int. Default value: 0x0.

ete.PIDR_REVISION

TRCPIDR REVISION value.

Type: int. Default value: 0x0.

ete.Q_CADENCE

Number of instruction blocks traced between two Q elements.

Type: int. Default value: 0x1.

ete.RES0_STATEFUL

Whether RES0 bits are stateful or RAZ/WI.

Type: bool. Default value: 0x0.

ete.RETSTACK

Return stack depth.

Type: `int`. Default value: `0x3`.

ete.REVISION

TRCIDR1 revision value.

Type: `int`. Default value: `0x0`.

ete.SIM_OVERFLOW_GRANULARITY

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

Type: `int`. Default value: `0x64`.

ete.SIM_OVERFLOW_PERCENTAGE

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

Type: `int`. Default value: `0x0`.

ete.SOURCE_ADDRESS

Allow generation of source address elements.

Type: `bool`. Default value: `0x0`.

ete.TRACE_OUTPUT

File to which to write trace byte stream.

Type: `string`. Default value: `""`.

ete.TRCSRSTA_FORCED_EXCEP

TRCSRSTA value for a forcibly traced exception.

Type: `bool`. Default value: `0x0`.

ext_abort_normal_noncacheable_read_is_sync

Synchronous reporting of normal noncacheable-read external aborts.

Type: `bool`. Default value: `0x1`.

ext_abort_so_write_ras_type

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.

Type: `int`. Default value: `0x2`.

force_mte_tag_access_razwi_and_ignore_tag_checks

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.

Type: `bool`. Default value: `0x0`.

force_zero_mpam_partid_and_pmg

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.

Type: `int`. Default value: `0x0`.

force_zero_PSTATE_PAN

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.

Type: `bool`. Default value: `0x0`.

GICDISABLE

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`. Default value: `0x1`.

has_coherent_icache

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`. Default value: `0x1`.

has_enhanced_pan

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.

Type: `int`. Default value: `0x2`.

has_ete

If true, implements the Embedded Trace Extension (FEAT_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

Type: `bool`. Default value: `0x0`.

has_external_rndr

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.

Type: `int`. Default value: `0x1`.

has_large_va

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.

Type: `int`. Default value: `0x0`.

has_peripheral_port

If true, an additional AXI peripheral port is configured.

Type: `bool`. Default value: `0x0`.

has_rndr

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT_RNG).

Type: `int`. Default value: `0x1`.

icache-hit_latency

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-maintenance_latency

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `int`. Default value: `0x0`.

icache-miss_latency

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`.

Type: `int`. Default value: `0x0`.

icache-prefetch_enabled

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

Type: `bool`. Default value: `0x0`.

icache-read_access_latency

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`.

Type: `int`. Default value: `0x0`.

icache-read_latency

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`.

Type: `int`. Default value: `0x0`.

icache-size

L1 I-Cache size in bytes.

Type: `int`. Default value: `0x8000`.

icache-state_modelled

Set whether I-cache has stateful implementation.

Type: `bool`. Default value: `0x0`.

instruction_tlb_size

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

Type: `int`. Default value: `0x0`.

memory_tagging_support_level

Equivalent to BROADCASTMTE.

Type: `int`. Default value: `0x3`.

mpam_max_partid

MPAM Maximum PARTID Supported.

Type: `int`. Default value: `0x1fff`.

mpam_max_vpmr

MPAM Maximum VPMR Supported.

Type: `int`. Default value: `0x7`.

NUM_CORES

Number of cores per cluster.

Type: `int`. Default value: `0x1`.

pmu-num_counters

Number of PMU counters implemented.

Type: `int`. Default value: `0x6`.

ptw_latency

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

Type: `int`. Default value: `0x0`.

rndr_rndrrs_seed

Initial seed for random engine used in RNDR register.

Type: `int`. Default value: `0x0`.

stage12_tlb_size

Number of stage1+2 tlb entries.

Type: `int`. Default value: `0x80`.

tlb_latency

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

Type: `int`. Default value: `0x0`.

trace_physical_registers_when_host_virtualisation_enabled

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.

Type: `int`. Default value: `0x1`.

treat_PAC_as_NOP

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.

Type: `bool`. Default value: `0x0`.

walk_cache_latency

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

Type: `int`. Default value: `0x0`.

4.5.71 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 4-220: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMSC000CT contains the following CADI targets:

- ARM_SC000

ARMSC000CT contains the following MTI components:

- [ARM_SC000](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)

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 4-221: 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.

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.
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_SC000

BIGENDINIT

Initialize processor to big endian mode.
Type: bool. Default value: 0x0.

BKPT

Number of breakpoint unit comparators implemented.
Type: int. Default value: 0x4.

cpi_div

divider for calculating CPI (Cycles Per Instruction).
Type: int. Default value: 0x1.

cpi_mul

multiplier for calculating CPI (Cycles Per Instruction).
Type: int. Default value: 0x1.

DBG

Set whether debug extensions are implemented.
Type: bool. Default value: 0x1.

IOP

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.
Type: bool. Default value: 0x0.

IRQDIS

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n].

Type: `int`. Default value: `0x0`.

master_id

Master ID presented in bus transactions.

Type: `int`. Default value: `0x0`.

min_sync_level

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

Type: `int`. Default value: `0x0`.

NUM_IRQ

Number of user interrupts.

Type: `int`. Default value: `0x20`.

NUM_MPU_REGION

Number of MPU regions.

Type: `int`. Default value: `0x0`.

scheduler_mode

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, 3=ISSCompare) WARNING:

This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `int`. Default value: `0x0`.

semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x20700000`.

semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

semihosting-stack_base

Virtual address of base of descending stack.

Type: `int`. Default value: `0x20800000`.**semihosting-stack_limit**

Virtual address of stack limit.

Type: `int`. Default value: `0x20700000`.**semihosting-Thumb_SVC**

T32 SVC number for semihosting.

Type: `int`. Default value: `0xab`.**SYST**

Enable support for SysTick timer functionality.

Type: `bool`. Default value: `0x1`.**use_architectural_names**

Enable using architectural names for registers LR/PSP/MSP instead of model-specific names.

Type: `bool`. Default value: `0x0`.**USER**

Enable support for Unprivileged/Privileged Extension.

Type: `bool`. Default value: `0x1`.**VTOR**

Include Vector Table Offset Register.

Type: `bool`. Default value: `0x1`.**WIC**

Include support for WIC-mode deep sleep.

Type: `bool`. Default value: `0x1`.**WPT**

Number of watchpoint unit comparators implemented.

Type: `int`. Default value: `0x2`.

4.5.72 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 4-222: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ARMSC300CT contains the following CADI targets:

- ARM_SC300

ARMSC300CT contains the following MTI components:

- [ARM_SC300](#)
- [AsyncCacheFlushUnit](#)
- [PVBusLogger](#)
- [PVBusMapper](#)

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 4-223: 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 4-224: Ports

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.

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.
currpri	Value	Master	Current execution priority.
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr[240]	Signal	Slave	This signal array delivers signals to the NVIC.
intnmi	Signal	Slave	Configure non maskable interrupt.
lockup	Signal	Master	Asserted when the processor is in lockup state.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
stcalib	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
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_SC300

BB_PRESENT

Enable bitbanding.
Type: bool. Default value: 0x1.

BIGENDINIT

Initialize processor to big endian mode.
Type: bool. Default value: 0x0.

cpi_div

divider for calculating CPI (Cycles Per Instruction).
Type: int. Default value: 0x1.

cpi_mul

multiplier for calculating CPI (Cycles Per Instruction).
Type: int. Default value: 0x1.

DBG_LVL

0: No debug; 1: Minimal debug; 2: Full debug without DWT address matching; 3: Full debug support with DWT data-comparators.
Type: int. Default value: 0x3.

LVL_WIDTH

Number of bits of interrupt priority.

Type: `int`. Default value: `0x3`.

master_id

Master ID presented in bus transactions.

Type: `int`. Default value: `0x0`.

min_sync_level

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

Type: `int`. Default value: `0x0`.

NUM_IRQ

Number of user interrupts.

Type: `int`. Default value: `0x10`.

NUM_MPU_REGION

Number of MPU regions.

Type: `int`. Default value: `0x8`.

scheduler_mode

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, 3=ISSCompare) WARNING:

This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `int`. Default value: `0x0`.

semihosting-cmd_line

Command line available to semihosting SVC calls.

Type: `string`. Default value: `""`.

semihosting-cwd

Base directory for semihosting file access.

Type: `string`. Default value: `""`.

semihosting-enable

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`. Default value: `0x1`.

semihosting-heap_base

Virtual address of heap base.

Type: `int`. Default value: `0x0`.

semihosting-heap_limit

Virtual address of top of heap.

Type: `int`. Default value: `0x20700000`.

semihosting-prefix

Prefix semihosting output with target instance name.

Type: `bool`. Default value: `0x0`.

semihosting-stack_base
Virtual address of base of descending stack.
Type: `int`. Default value: `0x20800000`.

semihosting-stack_limit
Virtual address of stack limit.
Type: `int`. Default value: `0x20700000`.

semihosting-Thumb_SVC
T32 SVC number for semihosting.
Type: `int`. Default value: `0xab`.

TRACE_LVL
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.
Type: `int`. Default value: `0x1`.

use_architectural_names
Enable using architectural names for registers LR/PSP/MSP instead of model-specific names.
Type: `bool`. Default value: `0x0`.

WIC
Include support for WIC-mode deep sleep.
Type: `bool`. Default value: `0x1`.

4.6 Media components

This section describes the Media components.

4.6.1 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 4-225: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

D71 contains the following CADI targets:

- [D71](#)

D71 contains the following MTI components:

- [D71](#)

- [PVBusSlave](#)

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 4-226: Ports

Name	Protocol	Type	Description
<code>apb_pvbus_s_adu</code>	PVBus	Slave	Slave port for register access.
<code>apb_pvbus_s_dpu</code>	PVBus	Slave	-
<code>axi_pvbus_adu_m</code>	PVBus	Master	Master AXI port for the AFBC unit
<code>axi_pvbus_lpu_m[2]</code>	PVBus	Master	Master AXI ports for pipelines
<code>display[2]</code>	LCD	Master	LCD ports for display outputs
<code>irq0_gcu_out</code>	Signal	Master	Shared interrupt owned by the GCU
<code>irq1_adu_out</code>	Signal	Master	Interrupt signal for the ADU block
<code>pixelclock_in[2]</code>	ClockSignal	Slave	Pixel clock inputs for the display outputs
<code>pvbus_tbu_m[2]</code>	PVBus	Master	Master ports for connection to TBU (SMMUv3)
<code>pvbus_tbu_s[2]</code>	PVBus	Slave	Slave ports for loopback from TBU (SMMUv3)
<code>reset_signal</code>	Signal	Slave	Reset signal.

4.6.2 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 4-227: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

DP500 contains the following CADI targets:

- DP500

DP500 contains the following MTI components:

- [DP500](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

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.

The model has the following 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 4-228: 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.

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Bus for processor 0.
pvbus_s	PVBus	Slave	Slave port for register access.
reset_signal	Signal	Slave	Reset signal.

4.6.3 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 4-229: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

DP500x2 contains the following CADI targets:

- DP500
- DP500x2

DP500x2 contains the following MTI components:

- [DP500](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

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.

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.

The model has the following 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 4-230: 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

Disable DMA snooping.

Type: bool. Default value: 0x0.

force_frame_rate

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.

Type: int. Default value: 0x0.

4.6.4 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 4-231: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

DP550 contains the following CADI targets:

- DP550

DP550 contains the following MTI components:

- [DP550](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

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.

The model has the following 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 4-232: 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.

Name	Protocol	Type	Description
reset_signal	Signal	Slave	Reset signal.

4.6.5 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 4-233: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

DP550x2 contains the following CADI targets:

- DP550
- DP550x2

DP550x2 contains the following MTI components:

- [DP550](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

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.

The model has the following 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 4-234: 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 DP550x2

disable_snooping_dma

Disable DMA snooping.

Type: bool. Default value: 0x0.

force_frame_rate

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.

Type: int. Default value: 0x0.

set_primary_nprot_nsaid

Configurable NPROT_NSaid for non-secure NSaid values(primary).

Type: int. Default value: 0xc00ba98.

set_primary_nprot_streamid

Configurable NPROT_STREAMID for non-secure STREAMID values(primary).

Type: int. Default value: 0xc00ba98.

set_primary_prot_nsaid

Configurable PROT_NSaid for secure NSaid values(primary).

Type: int. Default value: 0x4003210.

set_primary_prot_streamid

Configurable PROT_STREAMID for secure STREAMID values(primary).

Type: int. Default value: 0x4003210.

set_secondary_nprot_nsaid

Configurable NPROT_NSaid for non-secure NSaid values(secondary).

Type: int. Default value: 0xc00ba98.

set_secondary_nprot_streamid
Configurable NPROT_STREAMID for non-secure STREAMID values(secondary).
Type: int. Default value: 0xc00ba98.

set_secondary_prot_nsaaid
Configurable PROT_NSAID for secure NSAID values(secondary).
Type: int. Default value: 0x4003210.

set_secondary_prot_streamid
Configurable PROT_STREAMID for secure STREAMID values(secondary).
Type: int. Default value: 0x4003210.

4.6.6 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 4-235: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

DP650 contains the following CADI targets:

- DP650

DP650 contains the following MTI components:

- [DP650](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

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.

The model has the following 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 4-236: 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.

4.6.7 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 4-237: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

DP650x2 contains the following CADI targets:

- DP650
- DP650x2

DP650x2 contains the following MTI components:

- [DP650](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

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.

The model has the following 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 4-238: 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

Disable DMA snooping.

Type: bool. Default value: 0x0.

force_frame_rate

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.

Type: int. Default value: 0x0.

set_primary_nprot_nsaids

Configurable NPROT_NSaid for non-secure NSaid values(primary).

Type: int. Default value: 0xc00ba98.

set_primary_nprot_streamids

Configurable NPROT_STREAMID for non-secure STREAMID values(primary).

Type: int. Default value: 0xc00ba98.

set_primary_prot_nsaids

Configurable PROT_NSaid for secure NSaid values(primary).

Type: int. Default value: 0x4003210.

set_primary_prot_streamids

Configurable PROT_STREAMID for secure STREAMID values(primary).

Type: int. Default value: 0x4003210.

set_secondary_nprot_nsaids

Configurable NPROT_NSaid for non-secure NSaid values(secondary).

Type: int. Default value: 0xc00ba98.

set_secondary_nprot_streamids

Configurable NPROT_STREAMID for non-secure STREAMID values(secondary).

Type: int. Default value: 0xc00ba98.

set_secondary_prot_nsaId

Configurable PROT_NSAID for secure NSAID values(secondary).

Type: `int`. Default value: `0x4003210`.**set_secondary_prot_streamid**

Configurable PROT_STREAMID for secure STREAMID values(secondary).

Type: `int`. Default value: `0x4003210`.

4.6.8 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 4-239: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Mali_C5x_streaming_sink contains the following CADI targets:

- Mali_C5x_streaming_sink

Mali_C5x_streaming_sink contains the following MTI components:

- [PVBUSSlave](#)

Ports for Mali_C5x_streaming_sink

Table 4-240: 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

Saving captured frames on/off.

Type: `bool`. Default value: `0x1`.**fn_prefix**

Saved filenames prefix.

Type: `string`. Default value: `"isp_out_"`.

4.6.9 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 4-241: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Mali_C7x_streaming_sink contains the following CADI targets:

- Mali_C7x_streaming_sink

Mali_C7x_streaming_sink contains the following MTI components:

- [PVBusSlave](#)

Ports for Mali_C7x_streaming_sink

Table 4-242: 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

Saving captured frames on/off.

Type: `bool`. Default value: `0x1`.

fn_prefix

Saved filenames prefix.

Type: `string`. Default value: `"isp_out_"`.

4.6.10 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 4-243: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Mali_C55 parameters

hw_config

Enable or disable optional features. These features are controlled by virtual registers, any number of which can be configured by this parameter, separated by spaces.

Type: `str`. Default value: `$FRSCALER_FITTED=1 $PONG_CONFIG_FITTED=1`.

The `hw_config` parameter supports the following virtual registers and assignments:

\$CNR_FITTED

0

CNR and its square/sqrt and the relative config space are statically removed from the ISP. The CNR and its square/sqrt config address space are not writeable and reading it back returns 0.

1

CNR and its square/sqrt are present in the design.

Default value: 1.

\$COMPRESSION_FITTED

0

Temper compression is statically removed from the ISP.

1

Temper compression is present in the design.

Default value: 1.

\$DSPPIPE_FITTED

0

Downscale pipe and the relative config space are statically removed from the ISP. The downscale pipe config address space is not writeable and reading it back returns 0.

1

Downscale pipe is present in the design.

Default value: 1.

\$FRSCALER_FITTED**0**

Full resolution pipe scaler and the relative config space are statically removed from the ISP. The full resolution pipe scaler config address space is not writeable and reading it back returns 0.

1

Full resolution pipe scaler is present in the design.

Default value: 0.

\$IRIDIX_GTM_FITTED**0**

Iridix global tone mapping is removed from the ISP.

1

Iridix global tone mapping is present in the design.

Default value: 0.

\$IRIDIX_LTM_FITTED**0**

Iridix local tone mapping is removed from the ISP.

1

Iridix local tone mapping is present in the design.

Default value: 1.

\$PONG_CONFIG_FITTED**0**

Pong configuration space, with the exception of pong statistics memories, is statically removed from the ISP. The address space relative to the pong configuration space is not writeable and reading back the address space returns 0.

1

Pong configuration space is present in the design.

Default value: 1.

\$SCALER_COEF_SETS

Number of scaler coefficient sets available. This value is expressed in decimal.

Default value: 8.

\$SINTER_FITTED**0**

Sinter and the relative config space are statically removed from the ISP. The sinter config address space is not writeable and reading it back returns 0.

1

Sinter is present in the design.

Default value: 1.

\$SINTER_LITE**0**

Full sinter version used.

1

Sinter lite version used.

Default value: 0.

\$TEMPER_FITTED**0**

Temper and the relative config space are statically removed from the ISP. The temper config address space is not writeable and reading it back returns 0.

1

Temper is present in the design.

Default value: 1.

\$WDR_FITTED**0**

WDR Frame Stitch, offset and gain and the relative config registers are statically removed from the ISP.

1

WDR Frame Stitch, offset and gain are present in the design.

Default value: 1.

Limitations

The Mali_C55 ISP model has the following 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 4-244: Ports

Name	Protocol	Type	Description
ds_data_out_m[3]	PVBus	Master	-
ds_hsync_out_m	Signal	Master	-
ds_uv_valid_out_m	Signal	Master	-
ds_vsync_out_m	Signal	Master	-
fr_data_out_m[3]	PVBus	Master	Metadata + pixels, 16 bit, (RGB or YUV)
fr_hsync_out_m	Signal	Master	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
fr_uv_valid_out_m	Signal	Master	UV valid (set if both U and V pixels are valid)
fr_vsync_out_m	Signal	Master	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)
irq	Signal	Master	Shared interrupt
pvbus_m	PVBus	Master	Master AXI port for RAM access
pvbus_s	PVBus	Slave	Slave port for register access
reset_s	Signal	Slave	Reset signal
stream_data_in_s[3]	PVBus	Slave	Metadata + pixels, 20 bit
stream_hsync_in_s	Signal	Slave	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
stream_vsync_in_s	Signal	Slave	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)

4.6.11 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 4-245: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Model limitations

The Mali_C71 ISP model has the following limitations:

- Streaming input and output interfaces are supported but the protocol differs slightly from the real hardware.
- Fault Interface is not supported.
- Testing interrupt connectivity through the Interrupt Trigger register at offset 0x118DC is not supported.
- Double interrupt signal is 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_C71

Table 4-246: Ports

Name	Protocol	Type	Description
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)

4.6.12 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 4-247: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Model limitations

The Mali_C78 ISP model has the following limitations:

- Streaming input and output interfaces are supported but the protocol differs slightly from the real hardware.
- Fault Interface is not supported.
- Testing interrupt connectivity through the Interrupt Trigger register at offset 0x118Dc is not supported.
- Double interrupt signal is 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_C78

Table 4-248: Ports

Name	Protocol	Type	Description
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)

4.6.13 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 4-249: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Mali_Cxx_streaming_camera contains the following CADI targets:

- Mali_Cxx_streaming_camera

Mali_Cxx_streaming_camera contains the following MTI components:

- PVBusMaster
- PVBusSlave

Ports for Mali_Cxx_streaming_camera

Table 4-250: 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

A file containing one or more frames to stream in the ISP model's FRM format.

Type: `string`. Default value: `""`.

4.6.14 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 4-251: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Mali_G51 contains the following CADI targets:

- Mali_G51

Mali_G51 contains the following MTI components:

- [Mali_G51](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

About Mali_G51

The model has the following limitations:

- It does not support Arm®v8-style page tables.
- It does not execute GPU shader programs.
- It 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.
- 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 4-252: 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.

4.6.15 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 4-253: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Mali_G71 contains the following CADI targets:

- Mali_G71

Mali_G71 contains the following MTI components:

- [Mali_G71](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

About Mali_G71

The model has the following limitations:

- It does not support Arm®v8-style page tables.
- It does not execute GPU shader programs.
- It 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.
- 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 4-254: 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.

4.6.16 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 4-255: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Mali_G72 contains the following CADI targets:

- Mali_G72

Mali_G72 contains the following MTI components:

- [Mali_G72](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

About Mali_G72

The model has the following limitations:

- It does not support Arm®v8-style page tables.
- It does not execute GPU shader programs.
- It 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.
- 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 4-256: 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 pvbust_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.

4.6.17 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 4-257: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Mali_G76 contains the following CADI targets:

- Mali_G76

Mali_G76 contains the following MTI components:

- [Mali_G76](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

About Mali_G76

The model has the following limitations:

- It does not support Arm®v8-style page tables.
- It does not execute GPU shader programs.
- It 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 4-258: 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 pvbush_m port.
pvbush_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.

4.6.18 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 4-259: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Mali_G710 contains the following CADI targets:

- Mali_G710

Mali_G710 contains the following MTI components:

- [Mali_G710](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

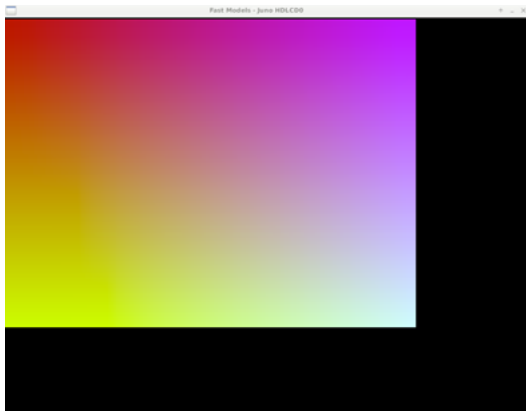
About Mali_G710 and Mali_G715

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, similar to the following:

Figure 4-2: Example frame from Mali_G710

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.

In addition to the default gradient pattern, these models also support delivering the output of real graphics workloads into the simulated system. This can be configured by offloading the rendering work to the host machine, through the GGA system of Fast Models.

For more information on configuring the Mali G710 and G715 models for host-rendered graphics, see [Graphics Acceleration in Fast Models](#) in *Fast Models User Guide*.

Configuring the Mali™ 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:

Table 4-260: Mali™ driver runtime settings

Parameter name	Type	Recommended value	Description
csf_firmware_boot_timeout_ms	Kernel module parameter	10000	Overrides the minimum timeout value for loading firmware into the model.
reset_timeout	Sysfs parameter	1000000	Overrides the minimum timeout when waiting for operations on the GPU to finish.
fw_timeout	Sysfs parameter	1000000	Overrides the minimum timeout when waiting for operations on the GPU to finish.

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 4-261: 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.

4.6.19 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 4-262: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Mali_G715 contains the following CADI targets:

- Mali_Turse

Mali_G715 contains the following MTI components:

- [Mali_Turse](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

About Mali_G710 and Mali_G715

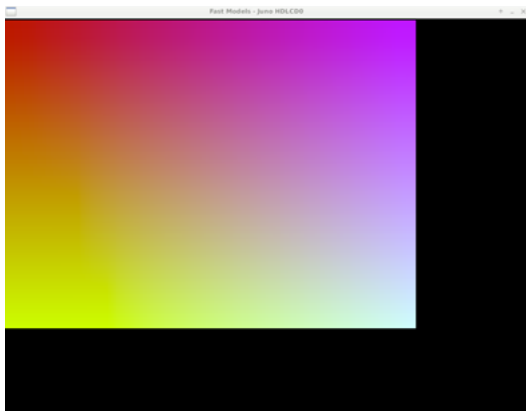
The Mali_G710 and Mali_G715 components model the Arm® Mali™ G710 and G715 GPUs respectively, which implement the Valhall architecture. They are among the first Mali™ GPUs to implement a Command Stream Frontend (CSF), which is a combination of hardware and firmware that helps to offload work from the 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, similar to the following:

Figure 4-3: Example frame from Mali_G710



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.

In addition to the default gradient pattern, these models also support delivering the output of real graphics workloads into the simulated system. This can be configured by offloading the rendering work to the host machine, through the GGA system of Fast Models.

For more information on configuring the Mali G710 and G715 models for host-rendered graphics, see [Graphics Acceleration in Fast Models](#) in *Fast Models User Guide*.

Configuring the Mali™ 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:

Table 4-263: Mali™ driver runtime settings

Parameter name	Type	Recommended value	Description
csf_firmware_boot_timeout_ms	Kernel module parameter	10000	Overrides the minimum timeout value for loading firmware into the model.
reset_timeout	Sysfs parameter	1000000	Overrides the minimum timeout when waiting for operations on the GPU to finish.
fw_timeout	Sysfs parameter	1000000	Overrides the minimum timeout when waiting for operations on the GPU to finish.

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 4-264: 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.

4.6.20 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 4-265: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Mali_G720 contains the following CADI targets:

- Mali_G720

Mali_G720 contains the following MTI components:

- [Mali_G720](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

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.

Ports for Mali_G720

Table 4-266: 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.

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.

Related information

[Deprecation of GGA](#)

4.6.21 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 4-267: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Mali_T624 contains the following CADI targets:

- Mali_T624

Mali_T624 contains the following MTI components:

- [Mali_T624](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

Ports for Mali_T624

Table 4-268: 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.

4.6.22 V61

Arm(r) Mali(tm)-V61 Video Processor. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-269: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

V61 contains the following CADI targets:

- V61

V61 contains the following MTI components:

- [PVBusMapper](#)
- [PVBusSlave](#)
- [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.

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`.

The model has the following 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 4-270: Ports**

Name	Protocol	Type	Description
<code>apb_s</code>	PVBus	Slave	APB Slave port for register access.
<code>axi_m</code>	PVBus	Master	AXI master bus for memory accesses.
<code>clk</code>	ClockSignal	Slave	Master clock, typically 300MHz.
<code>irq</code>	Signal	Master	IRQ signal to host CPU.
<code>reset</code>	Signal	Slave	Reset signal.

4.6.23 V550

Arm(r) Mali(tm)-V550 Video Processor. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-271: IP revisions support

Revision	Quality level
<code>r0p0</code>	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

V550 contains the following CADI targets:

- V550

V550 contains the following MTI components:

- [PVBusMapper](#)
- [PVBusSlave](#)
- [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 provided by Arm in the Third-Party IP (TPIP) package for convenience. Refer to the TPIP package for more details on FFomaxIL.

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`.

The model has the following 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 4-272: 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.

Related information

[Fixed Virtual Platforms](#)

4.7 Peripheral components

This section describes the Peripheral components.

4.7.1 AudioOut_File

File based Audio Output for PL041_AACI. This model is written in LISA+.

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.

expects 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 4-273: 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

Filename.

Type: `string`. Default value: `""`.

4.7.2 AudioOut_SDL

SDL based Audio Output for PL041_AACI. This model is written in LISA+.

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 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 4-274: Ports

Name	Protocol	Type	Description
audio	AudioControl	Slave	Audio input for a connection to a component such as the PL041_AACI.

4.7.3 Base_PowerController

Base Platforms Power Controller. This model is written in LISA+.

Base_PowerController contains the following CADI targets:

- Base_PowerController
- ClockTimerThread
- ClockTimerThread64
- SchedulerThread
- SchedulerThreadEvent

Base_PowerController contains the following MTI components:

- [Base_PowerController](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

Ports for Base_PowerController

Table 4-275: Ports

Name	Protocol	Type	Description
cpuporeset[36]	Signal	Master	-
dbgnopwrdown[36]	Signal	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

Whether core number is reflected in Affinity1 instead of Affinity0.

Type: bool. Default value: 0x0.

CPU-affinities

Definition of which cores are attached to the control pins, as a comma separated list of affinity dotted quads.

Type: string. Default value: "0.0.0.0".

CPU-available-mask

One bit per entry in CPU-affinities list, set zero if a CPU is wired up but actually not available.

Type: int. Default value: -0x1.

enable_lock_step

If lock step is enabled, the number of available cores get reduced to half.

Type: bool. Default value: 0x0.

startup

Comma-separated list of cores (wildcards allowed) to be powered up at startup or system reset.

Type: string. Default value: "0.0.0.*".

use_in_cluster_ppu

Set this to true if base power controller is connected to V9 core where in-cluster PPU is used, false, otherwise.

Type: bool. Default value: 0x0.

use_pchannel_for_threads

Set this to true if the pchannel is connected to cpus with thread support.

Type: bool. Default value: 0x0.

4.7.4 DebugAccessPort

This model is written in C++.

Ports for DebugAccessPort

Table 4-276: Ports

Name	Protocol	Type	Description
ap_pvbus_m[2]	PVBus	Master	-
clock	ClockSignal	Slave	-
paddrdbg31	Signal	Master	-

4.7.5 DebugROM

This model is written in C++.

Ports for DebugROM

Table 4-277: Ports

Name	Protocol	Type	Description
paddrdbg31	Signal	Master	-
pvbus_s	PVBus	Slave	-

4.7.6 DualClusterSystemConfigurationBlock

Dual Cluster System Configuration Block. This model is written in LISA+.

DualClusterSystemConfigurationBlock contains the following CADI targets:

- DualClusterSystemConfigurationBlock

DualClusterSystemConfigurationBlock contains the following MTI components:

- [DualClusterSystemConfigurationBlock](#)
- [PVBusSlave](#)

Ports for DualClusterSystemConfigurationBlock

Table 4-278: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-
cluster0_cfgend[4]	Signal	Master	-
cluster0_cfggte[4]	Signal	Master	-
cluster0_clusterid	Value	Master	-
cluster0_corereset[4]	Signal	Master	-

Name	Protocol	Type	Description
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	-
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

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!.

Type: int. Default value: 0x1.

Cluster0IdOnPOReset

ClusterId for primary cluster (Cortex-A15) on power-on reset.

Type: int. Default value: 0x0.

Cluster1IdOnPOReset

ClusterId for secondary cluster (Cortex-A7) on power-on reset.

Type: int. Default value: 0x1.

DCS_AID

DCS_AID is the Auxiliary ID Register.
Type: `int`. Default value: `0x0`.

DCS_ID

The value returned by the DCS_ID register.
Type: `int`. Default value: `0x41120000`.

DCS_ID_BUILD_NUMBER

DCS_ID build number.
Type: `int`. Default value: `0x1`.

DCS_LEDS

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.
Type: `int`. Default value: `0x0`.

DCS_SW

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.
Type: `int`. Default value: `0x0`.

DCSCB_PERIPHBASE

PERIPHBASE.
Type: `int`. Default value: `0x1e000000`.

FlipVGICWiringForCluster0AndCluster1

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.
Type: `bool`. Default value: `0x0`.

INTGEN_INTS

Number of custom IRQs controlled by interrupt generator is $\text{INTGEN_INTS} * 32 + 32$.
Type: `int`. Default value: `0x3`.

NumberOfCoresInCluster0

The number of cores in the primary cluster.
Type: `int`. Default value: `0x0`.

NumberOfCoresInCluster1

The number of cores in the secondary cluster.
Type: `int`. Default value: `0x0`.

ResetValueOfDaughterUserSwitches

Reset value of the user switches on the daughterboard.
Type: `int`. Default value: `0x0`.

stop_on_sequence_id

If non-zero the sequence_id of the SW trace mechanism on which to halt the simulator.
Type: `int`. Default value: `0x0`.

4.7.7 DummyAPB

DummyAPB. This model is written in LISA+.

DummyAPB contains the following CADI targets:

- DummyAPB

DummyAPB contains the following MTI components:

- [PVBusSlave](#)

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 4-279: Ports

Name	Protocol	Type	Description
<code>pvbuss_s</code>	PVBus	Slave	Bus slave interface.

Parameters for DummyAPB

fail

Abort all accesses.

Type: `bool`. Default value: `0x0`.

failmsg

String to print when 'fail'=true and access occurred.

Type: string. Default value: "".

periph_framesize

Size of frame (4/64, indicating if ID is at xFD0 or xFFD0).

Type: int. Default value: -0x1.

periphid_24

24 hex digits for the 12 bytes of peripheral ID.

Type: string. Default value: "".

periphid_generic

Set of space-separated offset:value pairs for dwords of ID.

Type: string. Default value: "".

ram_generic

Set of space-separated offset:default pairs for writable dwords.

Type: string. Default value: "".

warn_once

Warn once for the invalid read and write access.

Type: bool. Default value: 0x1.

4.7.8 ElfLoader

ELF loader component. This model is written in LISA+.

ElfLoader contains the following CADI targets:

- ElfLoader

ElfLoader contains the following MTI components:

- [PVBUSMaster](#)

About ElfLoader

ElfLoader provides an alternative method of loading ELF files into the system. It can load files in any of the following formats, or in gzip-compressed versions of them:

- ELF.
- Motorola S-Record.

Ports for ElfLoader

Table 4-280: Ports

Name	Protocol	Type	Description
pvbuse_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

ELF file.

Type: `string`. Default value: `""`.

impdef_copy

DEPRECATED: Use `realm_copy` or `root_copy` parameters. load ELF file to implementation defined memory spaces, if load file is not specified.

Type: `bool`. Default value: `0x0`.

lfile

load file for large address mapping.

Type: `string`. Default value: `""`.

ns_copy

copy whole file to NS memory space.

Type: `bool`. Default value: `0x1`.

output_attributes_parameter_of_core

Encoding of various attributes on the bus.

Type: `string`. Default value: `"ExtendedID[54:39]=MPAM_PARTID,`

`ExtendedID[38]=MPAM_SP[0], ExtendedID[37]=MPAM_SP[1], UserFlags[31:16]=IMPDEF2"`.

realm_copy

load ELF file to REALM memory spaces, if load file is not specified.

Type: `bool`. Default value: `0x0`.

root_copy

load ELF file to ROOT memory spaces, if load file is not specified.

Type: `bool`. Default value: `0x0`.

4.7.9 FlashLoader

A device that can preload a gzipped image into flash at startup. This model is written in C++.

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 4-281: 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.

Name	Protocol	Type	Description
flash_device2	FlashLoaderPort	Master	Used to program a flash device.
flash_device3	FlashLoaderPort	Master	Used to program a flash device.

4.7.10 GICv3CommsLogger

Traces GICv3Comms activity. This model is written in LISA+.

GICv3CommsLogger contains the following CADI targets:

- [GICv3CommsLogger](#)

GICv3CommsLogger contains the following MTI components:

- [GICv3CommsLogger](#)

Ports for GICv3CommsLogger

Table 4-282: 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

Print tracing information to attached debugger in addition to via MTI.

Type: bool. Default value: 0x0.

4.7.11 GICv3CommsPVBus

GICv3 Component for conversion between GICv3Comms protocol and PVBus. This model is written in C++.

GICv3CommsPVBus contains the following CADI targets:

- [GICv3CommsPVBus](#)

GICv3CommsPVBus contains the following MTI components:

- [GICv3CommsPVBus](#)
- [PVBusSlave](#)

Ports for GICv3CommsPVBus

Table 4-283: Ports

Name	Protocol	Type	Description
axi_master_id_s[256]	Value	Slave	-

Name	Protocol	Type	Description
distributor_s[256]	GICv3Comms	Slave	-
pvbus_m	PVBus	Master	-
pvbus_s	PVBus	Slave	-

4.7.12 GICv3ProtocolChecker

GICv3 Component for command protocol checking. This model is written in C++.

GICv3ProtocolChecker contains the following CADI targets:

- [GICv3ProtocolChecker](#)

GICv3ProtocolChecker contains the following MTI components:

- [GICv3ProtocolChecker](#)

Ports for GICv3ProtocolChecker

Table 4-284: Ports

Name	Protocol	Type	Description
cpu_comms	GICv3Comms	Master	Master GICv3Comms port.
gicv3_comms	GICv3Comms	Slave	Slave GICv3Comms port.

4.7.13 GUIPoll

Component providing a real-time periodic callback for GUI refresh. This model is written in C++.

About GUIPoll

Arm intends this component to be used as a subcomponent inside a LISA-based visualization component.

You can configure the period at which the GUIPoll component runs in milliseconds of real time, not simulation time. The component produces a `gui_callback()` signal at approximately this period, even when the simulation is paused.

You must implement the slave side of the `gui_callback()` signal in a LISA-based visualization component. Use this event to invoke the `visualisation::poll()` method to keep the GUI updated.

Ports for GUIPoll

Table 4-285: Ports

Name	Protocol	Type	Description
gui_callback	GUIPollCallback	Master	Sends callback requests to the visualization component.

4.7.14 HostBridge

Host Socket Interface Component. This model is written in C++.

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.

For more information see [2.10 User mode networking](#) on page 51.

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
```

Ports for HostBridge

Table 4-286: Ports

Name	Protocol	Type	Description
eth	VirtualEthernet	Slave	-
state	ValueState_64	Slave	-

Parameters for HostBridge

hostbridge.interfaceName

Host Interface.

Type: string. Default value: "".

hostbridge.userNetOptions

Control options for UserNet TCP/IP (for internal use only, please do not use).

Type: string. Default value: "".

hostbridge.userNetPorts

Listening ports to expose in user-mode networking.

Type: string. Default value: "".

hostbridge.userNetSubnet

Virtual subnet for user-mode networking.

Type: `string`. Default value: `"172.20.51.0/24"`.

hostbridge.userNetworking

Enable user-mode networking.

Type: `bool`. Default value: `0x0`.

Related information

[Configuring the networking environment for Microsoft Windows](#) on page 53

[Configuring the networking environment for Linux](#) on page 59

[Fast Models User Guide](#)

4.7.15 HostSerialInterface

Component which provides access to the host serial interface. This model is written in LISA+.

HostSerialInterface contains the following CADI targets:

- HostSerialInterface

Ports for HostSerialInterface

Table 4-287: Ports

Name	Protocol	Type	Description
SerialData	SerialData	Slave	Serial data connection to export to host machine.

Parameters for HostSerialInterface**baud_rate**

Baud rate override.

Type: `int`. Default value: `0x0`.

device

HW device to use.

Type: `string`. Default value: `"/dev/ttyS0"`.

4.7.16 IntelStrataFlashJ3

Intel Strata Flash J3 LISA+ model. This model is written in LISA+.

IntelStrataFlashJ3 contains the following CADI targets:

- IntelStrataFlashJ3

IntelStrataFlashJ3 contains the following MTI components:

- [PVBusMapper](#)
- [PVBusSlave](#)

About IntelStrataFlashJ3

This component is an efficient implementation of a NOR flash memory type device, an Intel StrataFlash Memory (J3).

In normal usage, the device acts as *Read Only Memory* (ROM) whose contents can be determined either by programming using the flashloader port or by using standard flash programming software running on the model, such as the Arm® Firmware Suite.

The implementation of this component is approximately that of the Intel part in the VE development board. The component is effectively organized as a bank of two 16-bit Intel Flash components forming a 32-bit component that can be read or programmed in parallel. The component supports all hardware behavior except for:

- Protection register.
- Enhanced configuration register.
- Unique device identifier.
- One time programmable cells.
- Suspend/resume, which is silently ignored.
- Status interrupt line.

All block operations are atomic. This means that the status register state machine status bit always reads 1, ready.

In normal operation, this component has no user-visible registers, but you can read from it as if it is memory.

Programming it or changing the configuration requires a sequence of special write operations, see general flash programming documentation. The component supports Common Flash Interface query operations, which allow drivers to determine the properties of the flash memory.



Note

The model interprets all writes as requests to the programming state machine, and there are many state-machine states that do not support subsequent reads and return `0xdeaddead` for them. Therefore, when simulating a ROM, use the `trapwrite=true` option.

Use the `diagnostics` parameter to select the level of diagnostic output:

Level 0

None.

Level 1

Report probable driver error operations:

- Unaligned operations that fault.
- Accesses that the state machine does not expect.
- Transitions of the state machine to unknown states.

- Writes to locked blocks and illegal lock commands.

Level 2

Report unimplemented and therefore ignored operations, and log lock commands.

Level 3

Warn if a flash write attempts to set bits (the write works if `unphysical_writes=true`).

Level 4

Log every read and write.

Ports for IntelStrataFlashJ3**Table 4-288: Ports**

Name	Protocol	Type	Description
flashloader	FlashLoaderPort	Slave	Can be used to load code/data into the flash.
mem_port	PVDevice	Slave	PVDevice port that is connect PVBusSlave port.
pvbus	PVBus	Slave	This is where the address and the data comes from the bus.

Parameters for IntelStrataFlashJ3**diagnostics**

Diagnostic level.

Type: `int`. Default value: `0x0`.

enable_read_status_logic

Enables logic to handle the status register reads as per the '3 Volt Intel StrataFlash Memory' specification.

Type: `bool`. Default value: `0x0`.

model_blocklock

Model per-block locking and set all the blocks to locked state on reset.

Type: `bool`. Default value: `0x0`.

size

Memory Size.

Type: `int`. Default value: `0x40000`.

trapwrite

Generate abort on write.

Type: `bool`. Default value: `0x0`.

unphysical_writes

Writes to flash are overwrite not AND.

Type: `bool`. Default value: `0x1`.

Related information

[Intel Download Center, Intel StrataFlash Memory \(J3\) datasheet](#)

4.7.17 Interrupt_Router

Interrupt Router Registers. This model is written in C++.

Ports for Interrupt_Router

Table 4-289: 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	-

4.7.18 LS64TestingFIFO

FIFO peripheral supporting LS64 accesses for testing purposes. This model is written in LISA+.

LS64TestingFIFO contains the following CADI targets:

- LS64TestingFIFO

LS64TestingFIFO contains the following MTI components:

- [PVBusSlave](#)

About LS64TestingFIFO

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 4-290: Ports

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	Bus subordinate interface.

Parameters for LS64TestingFIFO

buffer_size

The number of 64-byte slots in the FIFO.

Type: `int`. Default value: `0x2`.

op_type

The operation performed on the 64-byte transaction data 0 - None, 1 - Bitwise Negate.

Type: `int`. Default value: `0x1`.

4.7.19 MemoryMappedGenericTimer

ARM Generic Timer. This model is written in LISA+.

MemoryMappedGenericTimer contains the following CADI targets:

- MemoryMappedGenericTimer

MemoryMappedGenericTimer contains the following MTI components:

- [PVBusSlave](#)

Ports for MemoryMappedGenericTimer

Table 4-291: Ports

Name	Protocol	Type	Description
cntpsirq[8]	Signal	Master	-
cntvalueb	CounterInterface	Slave	For connection to MemoryMappedCounterModule.
pvbus_base_s[8]	PVBus	Slave	-
pvbus_ctlbase_s	PVBus	Slave	-
pvbus_el0base_s[8]	PVBus	Slave	-
timer_reset	Signal	Slave	Reset.

Parameters for MemoryMappedGenericTimer

bypass_ctlbase

Bypass CNTBase Access Control. Enable if only timer frame feature is required without CNTBase access control.

Type: `bool`. Default value: `0x0`.

cntel0acr_implemented

A bit-field of 8 bits, where bit {n} enables CNTELOACR for timer frame {n}.

Type: `int`. Default value: `0x0`.

diagnostics

Diagnostics.

Type: `int`. Default value: `0x0`.

frame_security

Hard-wired/configurable security for frames (N/S/X, one character per timer frame).

Type: `string`. Default value: `""`.

num_timers

Number of timer frames.

Type: `int`. Default value: `0x1`.

4.7.20 MemoryMappedGenericWatchdog

ARM Generic Watchdog. This model is written in LISA+.

MemoryMappedGenericWatchdog contains the following CADI targets:

- MemoryMappedGenericWatchdog

MemoryMappedGenericWatchdog contains the following MTI components:

- [PVBUSlave](#)

About MemoryMappedGenericWatchdog

This is a high-level watchdog that generates two interrupts rather than an interrupt followed by a reset.

Ports for MemoryMappedGenericWatchdog

Table 4-292: Ports

Name	Protocol	Type	Description
cntvalueb	CounterInterface	Slave	For connection to MemoryMappedCounterModule.
ctl_pvbus_s	PVBus	Slave	Access to control frame.
ref_pvbus_s	PVBus	Slave	Access to refresh frame.
reset_in	Signal	Slave	Reset.
WS0	Signal	Master	Set when watchdog expired for the first time.
WS1	Signal	Master	Set when watchdog expired for the second time.

Parameters for MemoryMappedGenericWatchdog

arch_version

Architecture version. Available 0 and 1.
Type: `int`. Default value: `0x0`.

diagnostics

Diagnostics.
Type: `int`. Default value: `0x0`.

NONSECURE

Non-Secure.
Type: `bool`. Default value: `0x0`.

product_id

Product Identifier.
Type: `int`. Default value: `0x0`.

4.7.21 NonVolatileCounter

Trusted Non-Volatile Counter unit. This model is written in LISA+.

NonVolatileCounter contains the following CADI targets:

- NonVolatileCounter

NonVolatileCounter contains the following MTI components:

- [NonVolatileCounter](#)
- [PVBusSlave](#)

Ports for NonVolatileCounter

Table 4-293: Ports

Name	Protocol	Type	Description
<code>pvbuss_s</code>	PVBus	Slave	Bus slave interface.

Parameters for NonVolatileCounter

diagnostics

Diagnostics.
Type: `int`. Default value: `0x0`.

rst_non_tz_fw_cnt

Value of NON_TZ_FW_CNT at reset.
Type: `int`. Default value: `0x0`.

rst_tz_fw_cnt

Value of TZ_FW_CNT at reset.
Type: `int`. Default value: `0x0`.

secure

Instantiate model as Secure (1) or NS (0).
Type: `bool`. Default value: `0x1`.

version

Version of the model functionality. Valid values are `r0` and `r1`.
Type: `string`. Default value: `"r0"`.

4.7.22 PCIeATC

This model is written in C++.

PCIeATC contains the following CADI targets:

- `validation_atc`

PCIeATC contains the following MTI components:

- [PVBusMapper](#)
- [PVBusSlave](#)
- [atc](#)

Ports for PCIeATC

Table 4-294: Ports

Name	Protocol	Type	Description
<code>atc</code>	<code>PCIeATC_get_if</code>	Slave	-
<code>disable_PRI_and_set_RF</code>	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.
<code>identify</code>	SMMUv3AEMIdentifyProtocol	Master	The user has a chance to determine how the substreamid is extracted from the transactions received on <code>pvbus_s</code> by using this port. If it is unimplemented then the ATC will use the default policy identified in SMMUv3_FOR_PCIE.lisa <201601041554/>
<code>pvbus_id_routed_s</code>	PVBus	Slave	-
<code>pvbus_m</code>	PVBus	Master	-
<code>pvbus_s</code>	PVBus	Slave	-
<code>uprgi</code>	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.

4.7.23 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 4-295: 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	-

4.7.24 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+.

PS2Keyboard contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PS2Keyboard
- SchedulerThread
- SchedulerThreadEvent

About PS2Keyboard

This component translates a stream of key press information into appropriate PS/2 serial data. The key press data stream must be provided by another component such as a visualization component.

Ports for PS2Keyboard

Table 4-296: 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.

4.7.25 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+.

PS2Mouse contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PS2Mouse
- SchedulerThread
- SchedulerThreadEvent

About PS2Mouse

This component implements the PS/2 register interface of a PS/2 style mouse. The mouse movement and button press data must come from another component such as a visualization component.

Ports for PS2Mouse

Table 4-297: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock signal to rate at which PS2Data signals are generated.
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.

Related information

[Visualisation library](#) on page 65

4.7.26 PVBusGICv3Comms

GICv3 Component for conversion between GICv3Comms protocol and PVBus. This model is written in C++.

PVBusGICv3Comms contains the following CADI targets:

- PVBusGICv3Comms

PVBusGICv3Comms contains the following MTI components:

- [PVBusGICv3Comms](#)
- [PVBusSlave](#)

Ports for PVBusGICv3Comms

Table 4-298: 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	-

4.7.27 PVMetaDataController

A simulation-only (not in hardware) component that can service metadata requests for transactions on PVBus. This model is written in C++.

Changes in 11.24.4

Parameters added:

- `mte_tag_carveout_json`
- `mte_tag_carveout_json_file`

`PVMetaDataController` provides the tag storage for Memory Tagging Extension versions FEAT_MTE2, FEAT_MTE3, and any later versions. It must be enabled and placed upstream of a RAM device for the feature to work for transactions routed to that RAM device. It might also be used to enable other future architectural features.



To enable `PVMetaDataController`, set the `is_enabled` parameter to true.

Ports for PVMetaDataController

Table 4-299: Ports

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

4.7.28 PchannelListener

Provides a dummy PChannel device to accept all request. This model is written in C++.

Ports for PchannelListener

Table 4-300: Ports

Name	Protocol	Type	Description
dev_pchannel_s	PChannel	Slave	-

4.7.29 RAMDevice

RAM device, can be dynamic or static ram. This model is written in LISA+.

RAMDevice contains the following CADI targets:

- RAMDevice

RAMDevice contains the following MTI components:

- PVBusSlave
- RAMDevice

Changes in 11.24.4

Parameters added:

- read_latency
- write_latency



As a generic device, this component does not have a hardware revision code.

Ports for RAMDevice

Table 4-301: Ports

Name	Protocol	Type	Description
pvbus	PVBus	Slave	Bus slave interface.

Parameters for RAMDevice

enable_atomic_ops

Supports Atomic Operations.

Type: bool. Default value: 0x0.

fill1

Fill pattern 1, initialise memory at start of simulation with alternating fill1, fill2 pattern.

Type: `int`. Default value: `0xdfdfdfcf`.

fill2

Fill pattern 2, initialise memory at start of simulation with alternating fill1, fill2 pattern.
Type: `int`. Default value: `0xcdfdfdf`.

read_latency

Memory read latency (ps/byte).
Type: `int`. Default value: `0x0`.

size

Memory Size.
Type: `int`. Default value: `0x100000000`.

write_latency

Memory write latency (ps/byte).
Type: `int`. Default value: `0x0`.

4.7.30 ROM

Simple ROM device. This model is written in LISA+.

ROM contains the following CADI targets:

- ROM

ROM contains the following MTI components:

- [PVBusSlave](#)

Ports for ROM

Table 4-302: Ports

Name	Protocol	Type	Description
pvbus	PVBus	Slave	-

Parameters for ROM

abort_writes

Abort writes instead of ignoring them.
Type: `bool`. Default value: `0x0`.

log2_size

Log2 size (bytes) e.g. 20 is 1 MiB.
Type: `int`. Default value: `0x14`.

raw_image

Raw image file to load at init time.
Type: `string`. Default value: `""`.

4.7.31 RandomNumberGenerator

Random Number Generator unit. This model is written in LISA+.

RandomNumberGenerator contains the following CADI targets:

- RandomNumberGenerator

RandomNumberGenerator contains the following MTI components:

- [PVBusSlave](#)

Ports for RandomNumberGenerator

Table 4-303: Ports

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	Bus slave interface.
RNG_intr	Signal	Master	Interrupt output.

Parameters for RandomNumberGenerator

diagnostics

Diagnostics.

Type: `int`. Default value: `0x0`.

seed

Random number seed.

Type: `int`. Default value: `0x0`.

4.7.32 RealTimeLimiter

Real Time Limiter. This model is written in LISA+.

RealTimeLimiter contains the following CADI targets:

- ClockDivider
- RealTimeLimiter

RealTimeLimiter contains the following MTI components:

- [ClockDivider](#)

Ports for RealTimeLimiter

Table 4-304: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock input.

Parameters for RealTimeLimiter

ENABLE

Rate limit simulation.

Type: `bool`. Default value: `0x0`.

RELATIVE_SPEED

Rate limit to at most this percentage of real time (100: limit to wall clock rate).

Type: `int`. Default value: `0x64`.

4.7.33 RealtimeClockTimer

Host Time Based Timer Module for Generic Timers. This model is written in C++.

Ports for RealtimeClockTimer

Table 4-305: Ports

Name	Protocol	Type	Description
set_frequency	Value_64	Slave	-
timer_callback	TimerCallback64	Master	-
timer_control	TimerControl64	Slave	-

4.7.34 RemapDecoder

The component that provides support for dynamically remappable regions of memory. This model is written in LISA+.

RemapDecoder contains the following CADI targets:

- RemapDecoder
- TZSwitch

RemapDecoder contains the following MTI components:

- [PVBusMapper](#)

Ports for RemapDecoder

Table 4-306: 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 TZSwitch

- tzswitch_0.normal**
Normal Port.
Type: `int`. Default value: `0x2`.
- tzswitch_0.secure**
Secure Port.
Type: `int`. Default value: `0x1`.

4.7.35 RootKeyStorage

Trusted Root-Key Storage unit. This model is written in LISA+.

RootKeyStorage contains the following CADI targets:

- RootKeyStorage

RootKeyStorage contains the following MTI components:

- [PVBusSlave](#)

Ports for RootKeyStorage

Table 4-307: Ports

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	Bus slave interface.

Parameters for RootKeyStorage

- diagnostics**
Diagnostics.
Type: `int`. Default value: `0x0`.
- hw_unique_key**
Hardware Unique Key (128-bit, 4 hex words).
Type: `string`. Default value: `"00000000 00000000 00000000 00000000"`.
- hw_unique_key_hex**
Hardware Unique Key (128-bit, little-endian hex byte stream).
Type: `string`. Default value: `""`.
- private_key**
Private Endorsement Key (256-bit, 8 hex words).
Type: `string`. Default value: `"00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000"`.
- private_key_hex**
Private Key (256-bit, little-endian hex byte stream).
Type: `string`. Default value: `""`.

public_key

Public Key (256-bit, 8 hex words).

Type: `string`. Default value: "00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000".

public_key_hex

Public Key (256-bit, little-endian hex byte stream).

Type: `string`. Default value: "".

ss_key

Secret Symmetric Key (128-bit, 4 hex words).

Type: `string`. Default value: "00000000 00000000 00000000 00000000".

ss_key_hex

Secret Symmetric Key (128-bit, little-endian hex byte stream).

Type: `string`. Default value: "".

version

Version of the model functionality. Valid values are `r0` and `r1`.

Type: `string`. Default value: "`r1`".

4.7.36 SMMUv3TestEngine

Test Engine used for testing SMMUv3. This model is written in C++.

SMMUv3TestEngine contains the following CADI targets:

- SMMUv3TestEngine

SMMUv3TestEngine contains the following MTI components:

- [PVBusSlave](#)
- [SMMUv3TestEngine](#)

Ports for SMMUv3TestEngine

Table 4-308: Ports

Name	Protocol	Type	Description
<code>client_s</code>	PCIDevice2ClientProtocol	Slave	-
<code>clk_in</code>	ClockSignal	Slave	-
<code>identify</code>	SMMUv3AEMIdentifyProtocol	Slave	-
<code>pvbus_control_s</code>	PVBus	Slave	-
<code>pvbus_m[64]</code>	PVBus	Master	-
<code>reset_in</code>	Signal	Slave	-

4.7.37 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+.

STLBusGasket contains the following CADI targets:

- STLBusGasket

STLBusGasket contains the following MTI components:

- PVBusMapper

Ports for STLBusGasket

Table 4-309: Ports

Name	Protocol	Type	Description
pvbus_in	PVBus	Slave	-
pvbus_out	PVBus	Master	-

Parameters for STLBusGasket

function

Function: 0-none, 1-STL value-forcing.
Type: `int`. Default value: 0x0.

reg_base

Base Address of STL control regs (ADDR,VAL at offsets 0,4).
Type: `int`. Default value: 0xe001e820.

verbose

Verbosity : 0-none, 1-some.
Type: `int`. Default value: 0x0.

4.7.38 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 4-310: 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.

4.7.39 TelnetTerminal

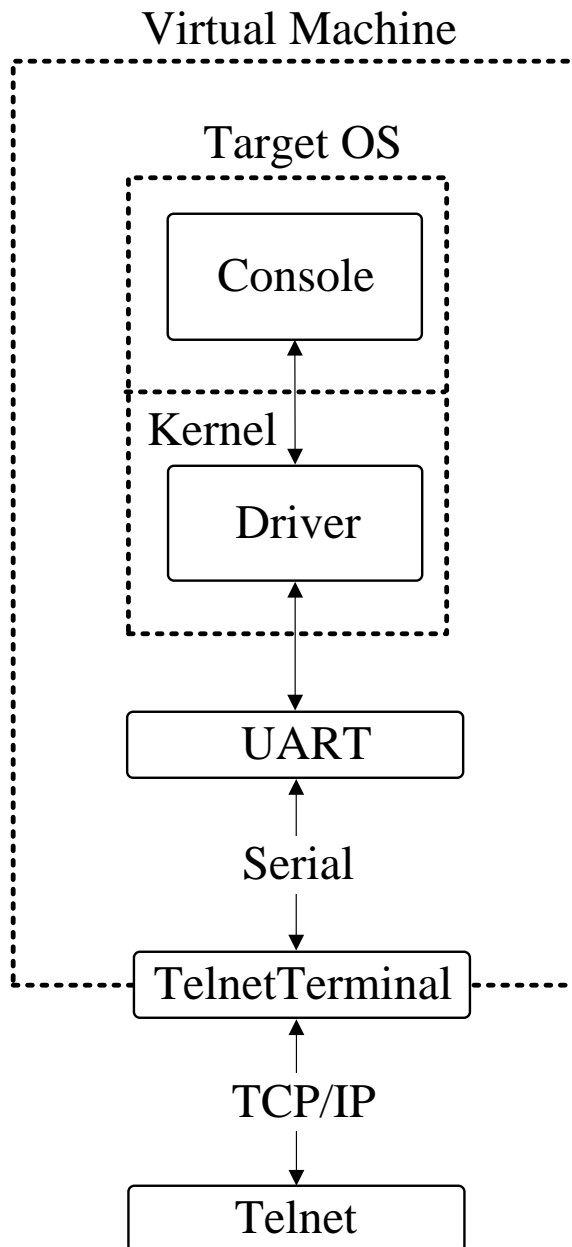
A host interface onto a serial port: exposes the two way serial data channel over a TCP/IP interface, and automatically opens a telnet application connected to the network socket, unless a user application connects first. This model is written in C++.

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 4-4: 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. 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 4-311: Ports

Name	Protocol	Type	Description
serial	SerialData	Slave	Slave port for connecting to a SerialData master.

Related information

[PL011_Uart](#) on page 1534

4.7.40 UnusedPrimeCell

A dummy component. It can be used to represent any unimplemented PrimeCell components. This model is written in LISA+.

UnusedPrimeCell contains the following CADI targets:

- UnusedPrimeCell

UnusedPrimeCell contains the following MTI components:

- [PVBusSlave](#)

Ports for UnusedPrimeCell

Table 4-312: Ports

Name	Protocol	Type	Description
pvbuss	PVBus	Slave	Bus slave interface.

4.7.41 VirtioBlockDevice

virtio block device. This model is written in C++.

VirtioBlockDevice contains the following CADI targets:

- [VirtioBlockDevice](#)

VirtioBlockDevice contains the following MTI components:

- [PVBusMaster](#)
- [PVBusSlave](#)
- [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 4-313: Ports

Name	Protocol	Type	Description
<code>intr</code>	Signal	Master	Virtio device sets interrupt to signal completion.
<code>pvbus</code>	PVBus	Slave	Virtio MMIO control/config/status registers.
<code>virtio_m</code>	PVBus	Master	Virtio device performs DMA accesses via master.

4.7.42 VirtioBlockDeviceMMIO

Virtio v1.0 Block device over MMIO transport. This model is written in C++.

VirtioBlockDeviceMMIO contains the following CADI targets:

- [VirtioBlockMMIO](#)

VirtioBlockDeviceMMIO contains the following MTI components:

- [PVBusMaster](#)

About VirtioBlockDeviceMMIO

VirtioBlockDeviceMMIO supports both the legacy and v1.0 OASIS virtio specifications.



VirtioBlockDeviceMMIO is an evolution of [4.7.41 VirtioBlockDevice](#) on page 1172, which is also MMIO-based and has the same ports, but only supports the legacy OASIS virtio specification.

Ports for VirtioBlockDeviceMMIO

Table 4-314: 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 VirtioBlockMMIO

enabled

Enable or disable device. If disabled, device can be accessed, but will not be activated.

Type: `bool`. Default value: `0x0`.

image_path

Image file path.

Type: `string`. Default value: `""`.

quiet

Don't print info or warnings (e.g. on malformed commands/descriptors).

Type: `bool`. Default value: `0x0`.

read_only

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.

Type: `bool`. Default value: `0x0`.

secure_accesses

Make device generate transactions with NS=0.

Type: `bool`. Default value: `0x0`.

target_endian

Target cpu endianness(0-LE, 1-BE).

Type: `bool`. Default value: `0x0`.

transaction_attributes

Transaction attributes used by device. 0x0 - inner-shared real access. 0x1 - outer-shared real access. 0x2 - outer-shared debug access.

Type: `int`. Default value: `0x0`.

transport

Choose legacy or modern virtio transport, if not specified, modern transport is used.

Type: `string`. Default value: `"modern"`.

4.7.43 VirtioNetMMIO

Virtio net device over MMIO transport. This model is written in C++.

VirtioNetMMIO contains the following CADI targets:

- HostBridge
- VirtioNetMMIO

VirtioNetMMIO contains the following MTI components:

- [PVBusMaster](#)

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
```

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 the related tasks at the end of this topic.

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>;
```

Ports for VirtioNetMMIO

Table 4-315: Ports

Name	Protocol	Type	Description
<code>intr</code>	Signal	Master	Virtio device sets interrupt to signal completion.
<code>pvbus</code>	PVBus	Slave	Virtio MMIO control/config/status registers.
<code>virtio_m</code>	PVBus	Master	Virtio device performs DMA accesses via master.

Parameters for HostBridge

`hostbridge.interfaceName`

Host Interface.

Type: string. Default value: "".

`hostbridge.userNetOptions`

Control options for UserNet TCP/IP (for internal use only, please do not use).

Type: string. Default value: "".

`hostbridge.userNetPorts`

Listening ports to expose in user-mode networking.

Type: string. Default value: "".

`hostbridge.userNetSubnet`

Virtual subnet for user-mode networking.

Type: `string`. Default value: `"172.20.51.0/24"`.

hostbridge.userNetworking

Enable user-mode networking.

Type: `bool`. Default value: `0x0`.

Parameters for VirtioNetMMIO**checksum**

For checksum-offloaded packets, if 'tx' is specified, outgoings will be checksummed by VirtioNet device; 'rx' is specified for incomings; 'all' for both.

Type: `string`. Default value: `""`.

enabled

Enable or disable device. If disabled, device can be accessed, but will not be activated.

Type: `bool`. Default value: `0x0`.

mac_address

Device MAC address, if not specified, a random MAC address is generated.

Type: `string`. Default value: `""`.

offload

Offload TCP/UDP segmentation/receiving operations to host.

Type: `string`. Default value: `"gso, gro"`.

secure_accesses

Make device generate transactions with NS=0.

Type: `bool`. Default value: `0x0`.

target_endian

Target cpu endianness(0-LE, 1-BE).

Type: `bool`. Default value: `0x0`.

transport

Choose legacy or modern virtio transport, if not specified, modern transport is used.

Type: `string`. Default value: `"modern"`.

Related information

[Configuring the networking environment for Microsoft Windows](#) on page 53

[Configuring the networking environment for Linux](#) on page 59

4.7.44 VirtioP9Device

virtio P9 server. This model is written in C++.

VirtioP9Device contains the following CADI targets:

- [VirtioP9Device](#)

VirtioP9Device contains the following MTI components:

- [PVBusMaster](#)

- [PVBusSlave](#)
- [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 4-316: 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.

4.7.45 VirtioPCIBlockDevice

virtio block device with PCI transport. This model is written in C++.

VirtioPCIBlockDevice contains the following CADI targets:

- [VirtioPCIBlockDevice](#)

VirtioPCIBlockDevice contains the following MTI components:

- [PVBusMaster](#)
- [PVBusSlave](#)
- [VirtioPCIBlockDevice](#)

About VirtioPCIBlockDevice

VirtioPCIBlockDevice is similar to VirtioBlockDevice, except it is PCI-based instead of MMIO-based. It supports the legacy OASIS virtio specification.

Ports for VirtioPCIBlockDevice

Table 4-317: 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.

4.7.46 VirtioRNG

virtio rng - Entropy device. This model is written in C++.

VirtioRNG contains the following CADI targets:

- VirtioEntropyMMIO

VirtioRNG contains the following MTI components:

- PVBusMaster

About VirtioRNG

VirtioRNG models a virtio entropy device as defined in the virtio 1.0 specification at <http://docs.oasis-open.org/virtio/virtio/v1.0/virtio-v1.0.html>. A virtual platform might need to integrate a VirtioRNG component to generate random numbers when:

- Linux or Android needs to generate kernel entropy. Hardware might do this using a timer, but this is not possible in the model because timers are not updated quickly enough.
- Security features are required, such as ssh.

Integrate VirtioRNG into a platform

Integrate the VirtioRNG component by instantiating it in your board's LISA file and connecting it to the SoC virtio master bus and interrupt signal as follows:

```
// Instantiate components
composition {
...
    virtio_rng : VirtioRNG();
...
}

connection {
...
    // Find a suitable address space and connect it to the SoC's virtio_m bus
    busdecoder.pvbus_m_range[0x001C190000..0x001C19ffff] => virtio_rng.pvbus;
    virtio_rng.virtio_m => self.virtio_m;

    // Connect the IRQ to the GIC IRQ
    virtio_rng.intr => gic400.irqs[101];
...
}
```

To configure Linux or Android for VirtioRNG, use the following build parameters:

Table 4-318: Linux and Android build parameters

Linux	Android
CONFIG_VIRTIO_MMIO=y	--enable CONFIG_VIRTIO_MMIO
CONFIG_HW_RANDOM=y	--enable CONFIG_HW_RANDOM
CONFIG_HW_RANDOM_VIRTIO=y	--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 4-319: Ports

Name	Protocol	Type	Description
intr	Signal	Master	Virtio device sets interrupt to signal completion.
pvbus	PVBus	Slave	Virtio control/config/status registers.
virtio_m	PVBus	Master	Virtio device performs DMA accesses via master.

Parameters for VirtioEntropyMMIO

diagnostics

Prints debug information: 0 = disabled; 1 = generated seed and device; 4 = generated seed, device and generated numbers.

Type: `int`. Default value: `0x0`.

enabled

Enable or disable device. If disabled, device can be accessed, but will not be activated.

Type: `bool`. Default value: `0x0`.

generator

User-defined generator: 0 = xorshiftstar; 1 = rand48; 2 = mersenne.

Type: `int`. Default value: `0x0`.

secure_accesses

Make device generate transactions with NS=0.

Type: `bool`. Default value: `0x0`.

seed

User-defined seed: 0 = uses a random seed; > 0 = user-defined fixed seed value.

Type: `int`. Default value: `0x0`.

target_endian

Target cpu endianness(O-LE, 1-BE).

Type: `bool`. Default value: `0x0`.

transport

Choose legacy or modern virtio transport, if not specified, modern transport is used.

Type: `string`. Default value: `"modern"`.

4.7.47 VirtualEthernetCrossover

Ethernet Crossover Cable. This model is written in LISA+.

VirtualEthernetCrossover contains the following CADI targets:

- VirtualEthernetCrossover

VirtualEthernetCrossover contains the following MTI components:

- [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 4-320: 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.

4.7.48 VirtualEthernetHub3

3 Port Ethernet Hub. This model is written in LISA+.

VirtualEthernetHub3 contains the following CADI targets:

- VirtualEthernetHub3

Ports for VirtualEthernetHub3

Table 4-321: 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.

4.7.49 VisEventRecorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions). This model is written in LISA+.

VisEventRecorder contains the following CADI targets:

- ClockDivider
- ClockTimerThread
- ClockTimerThread64
- SchedulerThread
- SchedulerThreadEvent
- VisEventRecorder

VisEventRecorder contains the following MTI components:

- [ClockDivider](#)

Ports for VisEventRecorder

Table 4-322: 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

check instruction count in recording file against actual instruction count during playback.
Type: `bool`. Default value: `0x1`.

playbackFileName

playback filename (empty string disables playback).
Type: `string`. Default value: `""`.

recordingFileName

recording filename (empty string disables recording).
Type: `string`. Default value: `""`.

recordingTimeBase

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).
Type: `int`. Default value: `0x5f5e100`.

verbose

enable verbose messages (1=normal, 2=even more).
Type: `int`. Default value: `0x0`.

4.7.50 WarningMemory

Memory that prints warnings, and RAZ/WIs or aborts. This model is written in C++.

Ports for WarningMemory

Table 4-323: Ports

Name	Protocol	Type	Description
pvbus	PVBus	Slave	Bus slave interface

4.7.51 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 Arm®v8-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 4-324: Ports

Name	Protocol	Type	Description
cti[36]	v8EmbeddedCrossTrigger_controlprotocol	Slave	-

4.8 Scheduler components

This section describes the Scheduler components.

4.8.1 AsyncSignal

Allows to cleanly schedule events from non-simulation threads onto the simulation thread. This model is written in C++.

Ports for AsyncSignal

Table 4-325: 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.

4.8.2 SchedulerInterface

A SchedulerInterface instance allows access to the Fast Models scheduler. This model is written in LISA+.

SchedulerInterface contains the following CADI targets:

- SchedulerInterface

Ports for SchedulerInterface

Table 4-326: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock frequency for waitTicks() function.
control	SchedulerInterfaceControl	Slave	Scheduler interface. Allows to: - wait for time

4.8.3 SchedulerThread

A SchedulerThread instance represents a co-routine thread in the simulation. This model is written in LISA+.

SchedulerThread contains the following CADI targets:

- SchedulerThread

Ports for SchedulerThread

Table 4-327: 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()

4.8.4 SchedulerThreadEvent

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on. This model is written in LISA+.

SchedulerThreadEvent contains the following CADI targets:

- SchedulerThreadEvent

Ports for SchedulerThreadEvent

Table 4-328: Ports

Name	Protocol	Type	Description
control	SchedulerThreadEventControl	Slave	SchedulerThreadEvent control. Masters use this to: - wait for this event - notify waiters that the event happened

4.9 Signals components

This section describes the Signals components.

4.9.1 AndGate

And Gate. This model is written in LISA+.

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 4-329: Ports

Name	Protocol	Type	Description
input[2]	Signal	Slave	2 input signals to be AND'ed.
output	Signal	Master	AND'ed output signal.

4.9.2 LabellerMasterIdExtendedIdUserFlag

Allows the modification of MasterID, ExtendedID and UserFlags attributes of PVBus transactions. This model is written in LISA+.

LabellerMasterIdExtendedIdUserFlag contains the following CADI targets:

- LabellerMasterIdExtendedIdUserFlag
- PVBusLogger

LabellerMasterIdExtendedIdUserFlag contains the following MTI components:

- [PVBusLogger](#)
- [PVBusMapper](#)

Ports for LabellerMasterIdExtendedIdUserFlag

Table 4-330: Ports

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

Parameters for LabellerMasterIdExtendedIdUserFlag

extendedid

ExtendedID value to be applied to transactions.

Type: `int`. Default value: `0x0`.

extendedid_mask

Mask used to determine which bits of extendedid parameter to be set in the transactions

ExtendedID attribute. `0xFFFFFFFFFFFFFFFF` will overwrite all the incoming ExtendedID bits with the value of the extendedid parameter, `0x0` will overwrite none.

Type: `int`. Default value: `0x0`.

masterid

MasterID value to be applied to transactions.

Type: `int`. Default value: `0x0`.

masterid_mask

Mask used to determine which bits of masterid parameter to be set in the transactions

MasterID attribute. `0xFFFFFFFF` will overwrite all the incoming MasterID bits with the value of the masterid parameter, `0x0` will overwrite none.

Type: `int`. Default value: `0x0`.

userflags

UserFlags value to be applied to transactions.

Type: `int`. Default value: `0x0`.

userflags_mask

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.

Type: `int`. Default value: `0x0`.

Parameters for PVBusLogger

pvbuslogger.trace_debug

Enable tracing of debug transactions.

Type: `bool`. Default value: `0x0`.

`pvbuslogger.trace_snoops`

Enable tracing of ACE snoop requests.

Type: `bool`. Default value: `0x0`.

4.9.3 LabellerUserSignals

This model is written in LISA+.

LabellerUserSignals contains the following CADI targets:

- LabellerUserSignals
- PVBusLogger

LabellerUserSignals contains the following MTI components:

- [PVBusLogger](#)
- [PVBusMapper](#)

Ports for LabellerUserSignals

Table 4-331: Ports

Name	Protocol	Type	Description
<code>pvbus_m</code>	PVBus	Master	Output with modified UserFlags.
<code>pvbus_s</code>	PVBus	Slave	Unmodified input.

Parameters for LabellerUserSignals

`user`

User signal to be applied to transactions.

Type: `int`. Default value: `0x0`.

Parameters for PVBusLogger

`pvbuslogger.trace_debug`

Enable tracing of debug transactions.

Type: `bool`. Default value: `0x0`.

`pvbuslogger.trace_snoops`

Enable tracing of ACE snoop requests.

Type: `bool`. Default value: `0x0`.

4.9.4 OrGate

Or Gate. This model is written in LISA+.

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 4-332: Ports

Name	Protocol	Type	Description
input[16]	Signal	Slave	16 input signals to be OR'ed.
output	Signal	Master	OR'ed output signal.

4.9.5 SGSignalBuffer

Buffer to synchronise SystemGenerator Signal setValue() calls. This model is written in LISA+.

SGSignalBuffer contains the following CADI targets:

- SGSignalBuffer



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

Ports for SGSignalBuffer

Table 4-333: 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.

4.9.6 SignalDriver

Drives signal port based on parameter, register or bus slave port. This model is written in LISA+.

SignalDriver contains the following CADI targets:

- SignalDriver

SignalDriver contains the following MTI components:

- PVBUSSlave
- SignalDriver

Ports for SignalDriver

Table 4-334: Ports

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	-
signal_out	Signal	Master	-

Parameters for SignalDriver

param_input

Drive signal_out port with this parameter value.

Type: bool. Default value: 0x0.

4.9.7 SignalLogger

Traces signal activity. This model is written in LISA+.

SignalLogger contains the following CADI targets:

- SignalLogger

SignalLogger contains the following MTI components:

- [SignalLogger](#)

Ports for SignalLogger

Table 4-335: Ports

Name	Protocol	Type	Description
in	Signal	Slave	Input signal port.
out	Signal	Master	Output signal port.

Parameters for SignalLogger

forward_signal

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

Type: bool. Default value: 0x1.

4.9.8 Value64Logger

Traces value activity. This model is written in LISA+.

Value64Logger contains the following CADI targets:

- Value64Logger

Value64Logger contains the following MTI components:

- [Value64Logger](#)

Ports for Value64Logger

Table 4-336: Ports

Name	Protocol	Type	Description
in	Value_64	Slave	Input signal port.
out	Value_64	Master	Output signal port.

4.9.9 ValueLogger

Traces value activity. This model is written in LISA+.

ValueLogger contains the following CADI targets:

- ValueLogger

ValueLogger contains the following MTI components:

- [ValueLogger](#)

Ports for ValueLogger

Table 4-337: Ports

Name	Protocol	Type	Description
in	Value	Slave	Input signal port.
out	Value	Master	Output signal port.

4.10 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.

4.10.1 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 4-338: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

AHCI_SATA contains the following CADI targets:

- AHCI_SATA

AHCI_SATA contains the following MTI components:

- [ExportTestAHCI_SATA](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

Ports for AHCI_SATA

Table 4-339: Ports

Name	Protocol	Type	Description
ahci_dma_m	PVBus	Master	-
client_s	PCIDevice2ClientProtocol	Slave	-
pvbust	PVBus	Slave	-

4.10.2 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 4-340: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

BP141_TZMA contains the following CADI targets:

- BP141_TZMA
- PVBusRange
- TZSwitch

BP141_TZMA contains the following MTI components:

- [PVBusMapper](#)

About BP141_TZMA

BP141_TZMA permits a single physical memory cell of up to 2MB 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 4KB increments up to a maximum of 2MB.

TZMEMSIZE is the maximum addressing range of the memory as defined by that parameter. By default, TZMEMSIZE is set to 2MB. In the following table, AxADDR is the offset address that the transactions want to access.

Table 4-341: 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 4-342: 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

Addressable range of device.

Type: `int`. Default value: `0x200000`.

TZSECROMSIZE

Default secure size.

Type: `int`. Default value: `0x200`.

TZSEGSIZE

Segment size.

Type: `int`. Default value: `0x1000`.

Parameters for PVBusRange

pvbusrange_0.range

Addressable range routed to pvbus_port_a.

Type: `int`. Default value: `0x0`.

Parameters for TZSwitch

tzswitch_0.normal

Normal Port.

Type: `int`. Default value: `0x2`.

tzswitch_0.secure

Secure Port.

Type: `int`. Default value: `0x1`.

4.10.3 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 4-343: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

BP147_TZPC contains the following CADI targets:

- BP147_TZPC

BP147_TZPC contains the following MTI components:

- [PVBusSlave](#)

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 4-344: 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.

Name	Protocol	Type	Description
TZPCR0SIZE	Value	Master	Output secure RAM region size.

4.10.4 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 4-345: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CCI400 contains the following CADI targets:

- CCI400

CCI400 contains the following MTI components:

- [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 Cortex®-A15 and the 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/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 4-346: Ports

Name	Protocol	Type	Description
acchannelen	Value	Slave	For each upstream port, determine if it is enabled or not with respect to snoop requests.

Name	Protocol	Type	Description
barrierterminate	Value	Slave	For each downstream port, determine if barriers are terminated at that port.
broadcastcachemain	Value	Slave	For each downstream port, determine if broadcast cache maintenance operations are forwarded down that port. A three bit signal but as the model only have a single downstream port, setting any of the bits will make it work.
bufferableoverride	Value	Slave	For each downstream port, determine if all transactions are forced to non-bufferable (AWCACHE[0] is forced to 0).
clk_in	ClockSignal	Slave	Clock signal for cciregisters
errorirq	Signal	Master	A signal stating that the imprecise error register is nonzero.
evntcntoverflow[5]	Signal	Master	When an event counter overflows, it sets the corresponding signal.
lint_ace_3_reset_state	Signal	Slave	This port can be connected to the reset signals of the system attached to the pvbus_s_ace_3 port.
lint_ace_4_reset_state	Signal	Slave	This port can be connected to the reset signals of the system attached to the pvbus_s_ace_4 port.
periphbase	Value_64	Slave	This port sets the base address of the private peripheral region.
pvbus_m	PVBus	Master	Master port for all downstream memory accesses.
pvbus_s_ace_3	PVBus	Slave	ACE-capable slave ports.
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

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.
Type: int. Default value: 0x1f.

barrierterminate

For each downstream port, determine if barriers will be terminated at that port.
Type: int. Default value: 0x7.

broadcastcachemain

For each downstream port a bit determines if broadcast cache maintenance operations are forwarded down that port.
Type: int. Default value: 0x0.

bufferableoverride

For each downstream port, determine if all transactions will be forced to non-bufferable.
Type: int. Default value: 0x0.

cache_state_modelled

Model the cache state.

Type: `bool`. Default value: `0x1`.

force_on_from_start

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'.

Type: `bool`. Default value: `0x0`.

is_downstream_domain_boundary_for_far_atomic

This interconnect is at the last stage of the domain boundary.

Type: `bool`. Default value: `0x0`.

log_enabled

Enable log messages from the CCI register file. Log level 0 means do not print anything, 1 means print only access violations, 2 means also print writes, 3 means print reads as well.

Type: `int`. Default value: `0x1`.

periphbase

Value for PERIPHBASE. Only bits [39:16] are used. This value may be overridden by an input on the periphbase port.

Type: `int`. Default value: `0x2c000000`.

revision

Revision of the CCI400.

Type: `string`. Default value: `"r0p0"`.

4.10.5 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 4-347: IP revisions support

Revision	Quality level
r0p0	Full support
r0p2	Full support
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CCI500 contains the following CADI targets:

- CCI500

CCI500 contains the following MTI components:

- [CCI500](#)
- [PVBusSlave](#)

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 `pvtbus_s[1]` is ACE, `pvtbus_s[0]` is ACE-Lite+DVM and `pvtbus_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 `SILVERR` 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 (`EVENTBUS`).

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 4-348: Ports

Name	Protocol	Type	Description
acchannelensx[7]	Value	Slave	ACCHANNLENSx represents the ports ACCHANNLENS0..ACCHANNLENS7 on the RTL (assuming there are seven upstream ports). * each upstream ACE port 'y' (pvbus_s[y]) has a two bit ACCHANNLENSx * 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 ACCHANNLENSx * bit 0 == 0 -- DVM messages are disabled from being sent to this interface In the model, as we support a variety of configurations with a single LISA file then each port will behave as though it is one bit or two bit as appropriate. If you send a value that cannot be represented, given the width of the port, then the CCI model will halt and produce a fatal error. The assumed values of these are set by parameters until they are driven, so you need not drive them if they are constant. In the RTL, these signals are sampled at reset. Due to ordering issues w.r.t. reset() on different components then we cannot do that. Instead the signals are sampled at first transaction. Thus any controller that is producing these signals has to hold them constant for long enough. AC channel enables.
address_decoder	CCI500_ AddressDecoderProtocol	Master	An address decoder can be attached to the address_decoder port to choose which pvbus_s port a downstream transaction will go out of. If you do not connect an address decoder then all transactions will go out of port 0.
dbgen	Signal	Slave	Invasive debug enable.
errirq	Signal	Master	Indicates that an error response, DECERR or SLVERR, is received on the RRESP, BRESP, or CRRESP input signals, and it cannot be signaled precisely.
evntcntoverflow[8]	Signal	Master	Overflow flags for the PMU clock and counters.
niden	Signal	Slave	Non-invasive debug enable.
pvbus_m[6]	PVBus	Master	Bus master ports.
pvbus_register_file_s	PVBus	Slave	The slave port of the register file.
pvbus_s[7]	PVBus	Slave	Bus slave ports.
reset_in	Signal	Slave	Reset the interconnect.
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

For upstream port 0 determine if it is enabled or not w.r.t. snoop requests.

Type: int. Default value: 0x0.

acchannelens1

For upstream port 1 determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0x0`.

acchannelens2

For upstream port 2 determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0x0`.

acchannelens3

For upstream port 3 determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0x0`.

acchannelens4

For upstream port 4 determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0x0`.

acchannelens5

For upstream port 5 determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0x0`.

acchannelens6

For upstream port 6 determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0x0`.

addr_width

The bit-width of the address that the CCI can accept.

Type: `int`. Default value: `0x28`.

cache_state_modelled

Model the cache state.

Type: `bool`. Default value: `0x1`.

dbgen

Invasive debug enable. If true, enables the counting of PMU events.

Type: `bool`. Default value: `0x1`.

enable_logger

Enable PVBUSLoggers for the downstream ports in the CCI model.

Type: `bool`. Default value: `0x0`.

force_on_from_start

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 disavow the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.

Type: `bool`. Default value: `0x0`.

niden

Whether non-secure events are allowed to be counted in the performance monitor.

Type: `bool`. Default value: `0x1`.

num_ace_lite_ports

The bottom `num_ace_lite_ports` are ACE-Lite+DVM.

Type: `int`. Default value: `0x5`.

num_ace_ports

The top `num_ace_ports` are ACE and support full coherency.

Type: `int`. Default value: `0x2`.

number_of_phantom_entries

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.

Type: `int`. Default value: `0x20`.

qos_threshold_upper

Reset value for the QoS threshold register.

Type: `int`. Default value: `0xc`.

reentrancy_support

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'.

Type: `string`. Default value: `"env"`.

spiden

Secure invasive debug enable. If both `SPIDEN` and `DBGEN` are high, enables the counting of both Non-secure and Secure events.

Type: `bool`. Default value: `0x1`.

spniden

Whether secure and non-secure events are allowed to be counted in the performance monitor.

Type: `bool`. Default value: `0x1`.

version

The version of the interconnect. Allowed versions are:- `r0p0`, `r0p2`, `r1p0`.

Type: `string`. Default value: `""`.

4.10.6 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 4-349: IP revisions support

Revision	Quality level
r0p0	Full support
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CCI550 contains the following CADI targets:

- CCI550

CCI550 contains the following MTI components:

- [CCI550](#)
- [PVBusSlave](#)

Differences between the model and the RTL

Address Decoder

- Only supports striping down to 4KiB.
- If the address decoder aborts the access, returns `SILVERR` 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 `pvbuss_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 4-350: Ports

Name	Protocol	Type	Description
acchannelensx[7]	Value	Slave	The acchannelensx[N] pins are used to tell the interconnect if the upstream system will accept snoops and/or DVM messages.
address_decoder	CCI500_ AddressDecoderProtocol	Master	An address decoder can be attached to the address_decoder port to choose which pvbus_s port a downstream transaction will go out of. If you do not connect an address decoder then all transactions will go out of port 0.
dbgen	Signal	Slave	Invasive debug enable.
errirq	Signal	Master	Some async error was detected.
evntcntoverflow[8]	Signal	Master	The output interrupts of the event counters.
niden	Signal	Slave	Non-invasive debug enable.
pvbus_m[7]	PVBus	Master	The downstream master ports.
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

For upstream port 0 determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0x0`.

acchannelens1

For upstream port 1 determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0x0`.

acchannelens2

For upstream port 2 determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0x0`.

acchannelens3

For upstream port 3 determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0x0`.

acchannelens4

For upstream port 4 determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0x0`.

acchannelens5

For upstream port 5 determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0x0`.

acchannelens6

For upstream port 6 determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0x0`.

addr_width

The bit-width of the address that the CCI can accept.

Type: `int`. Default value: `0x28`.

cache_state_modelled

Model the cache state.

Type: `bool`. Default value: `0x1`.

dbgen

Invasive debug enable. If true, enables the counting of PMU events.

Type: `bool`. Default value: `0x1`.

enable_logger

Enable PVBUSLoggers for the downstream ports in the CCI model.

Type: `bool`. Default value: `0x0`.

force_on_from_start

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 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 disavow the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.

Type: `bool`. Default value: `0x0`.

niden

Whether non-secure events are allowed to be counted in the performance monitor.

Type: `bool`. Default value: `0x1`.

num_ace_lite_ports

The bottom `num_ace_lite_ports` are ACE-Lite+DVM.

Type: `int`. Default value: `0x5`.

num_ace_ports

The top `num_ace_ports` are ACE and support full coherency.

Type: `int`. Default value: `0x2`.

number_of_phantom_entries

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.

Type: `int`. Default value: `0x20`.

qos_threshold_upper

Reset value for the QoS threshold register.

Type: `int`. Default value: `0xc`.

reentrancy_support

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'.

Type: `string`. Default value: `"env"`.

si0_qos_bw_regulator

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.

Type: `bool`. Default value: `0x0`.

si1_qos_bw_regulator

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.

Type: `bool`. Default value: `0x0`.

si2_qos_bw_regulator

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.

Type: `bool`. Default value: `0x0`.

si3_qos_bw_regulator

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.

Type: `bool`. Default value: `0x0`.

si4_qos_bw_regulator

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.

Type: `bool`. Default value: `0x0`.

si5_qos_bw_regulator

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.

Type: `bool`. Default value: `0x0`.

si6_qos_bw_regulator

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.

Type: `bool`. Default value: `0x0`.

si_system_coherency_interface

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 SCl port is connected but `si_system_coherency_interface` disable 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.

Type: `int`. Default value: `0x0`.

spiden

Secure invasive debug enable. If both SPIDEN and DBGGEN are high, enables the counting of both Non-secure and Secure events.

Type: `bool`. Default value: `0x1`.

spniden

Whether secure and non-secure events are allowed to be counted in the performance monitor.

Type: `bool`. Default value: `0x1`.

version

The version of the interconnect. Allowed versions are:- `r0p0`, `r1p0`.

Type: `string`. Default value: `""`.

4.10.7 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 4-351: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CCN502 contains the following CADI targets:

- CCN5XX
- CCNCache

CCN502 contains the following MTI components:

- [CCNCache](#)
- [CCNRegisterSet](#)
- [PVBusExclusiveMonitor](#)
- [PVBusMapper](#)
- [PVBusSlave](#)

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.

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 `ccn_cache` subcomponent are not accessible in System Canvas.

Ports for CCN502

Table 4-352: 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 CCNCache

`ccn_cache.acchannelen_rnf`

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0xf`.

`ccn_cache.acchannelen_rni`

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

Type: `int`. Default value: `0xffff`.

`ccn_cache.cache_size_kb`

Number of kilo bytes in cache.

Type: `int`. Default value: `0x2000`.

`ccn_cache.cache_state_modelled`

Model the cache state.

Type: `bool`. Default value: `0x1`.

`ccn_cache.force_on_from_start`

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'.

Type: `bool`. Default value: `0x0`.

`ccn_cache.number_of_phantom_entries`

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.

Type: `int`. Default value: `0x20`.

`ccn_cache.periphbase`

Value for PERIPHBASE. Only bits [43:24] are used.

Type: `int`. Default value: `0x2c000000`.

ccn_cache.reentrancy_support

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'.

Type: `string`. Default value: `"env"`.

ccn_cache.sbas_bridge_present

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

Type: `bool`. Default value: `0x1`.

ccn_cache.sbsx_bridge_present

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

Type: `bool`. Default value: `0x1`.

4.10.8 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 4-353: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CCN504 contains the following CADI targets:

- CCN5XX
- CCNCache

CCN504 contains the following MTI components:

- [CCNCache](#)
- [CCNRegisterSet](#)
- [PVBusExclusiveMonitor](#)
- [PVBusMapper](#)
- [PVBusSlave](#)

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 4-354: Ports

Name	Protocol	Type	Description
<code>pvbuss_m_hni[1]</code>	PVBus	Master	HNI downstream port.
<code>pvbuss_m_snf[2]</code>	PVBus	Master	SNF downstream ports.
<code>pvbuss_s_rnf[4]</code>	PVBus	Slave	RNF upstream ports.
<code>pvbuss_s_rni[18]</code>	PVBus	Slave	RNI upstream ports.
<code>reset_in</code>	Signal	Slave	Reset signal.

Parameters for CCNCache

`ccn_cache.acchannelen_rnf`

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0xf`.

`ccn_cache.acchannelen_rni`

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

Type: `int`. Default value: `0xffff`.

`ccn_cache.cache_size_kb`

Number of kilo bytes in cache.

Type: `int`. Default value: `0x2000`.

`ccn_cache.cache_state_modelled`

Model the cache state.

Type: `bool`. Default value: `0x1`.

`ccn_cache.force_on_from_start`

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'.

Type: `bool`. Default value: `0x0`.

`ccn_cache.number_of_phantom_entries`

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.

Type: `int`. Default value: `0x20`.

ccn_cache.periphbase

Value for PERIPHBASE. Only bits [43:24] are used.

Type: `int`. Default value: `0x2c000000`.

ccn_cache.reentrancy_support

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'.

Type: `string`. Default value: `"env"`.

ccn_cache.sbas_bridge_present

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

Type: `bool`. Default value: `0x1`.

ccn_cache.sbsx_bridge_present

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

Type: `bool`. Default value: `0x1`.

4.10.9 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 4-355: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CCN508 contains the following CADI targets:

- CCN5XX
- CCNCache

CCN508 contains the following MTI components:

- [CCNCache](#)
- [CCNRegisterSet](#)
- [PVBusExclusiveMonitor](#)
- [PVBusMapper](#)
- [PVBusSlave](#)

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 4-356: Ports

Name	Protocol	Type	Description
<code>pvbuss_m_hni[2]</code>	PVBus	Master	HNI downstream ports.
<code>pvbuss_m_snf[4]</code>	PVBus	Master	SNF downstream ports.
<code>pvbuss_s_rnf[8]</code>	PVBus	Slave	RNF upstream ports.
<code>pvbuss_s_rni[24]</code>	PVBus	Slave	RNI upstream ports.
<code>reset_in</code>	Signal	Slave	Reset signal.

Parameters for CCNCache

`ccn_cache.acchannelen_rnf`

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

Type: `int`. Default value: `0xf`.

`ccn_cache.acchannelen_rni`

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

Type: `int`. Default value: `0xffff`.

`ccn_cache.cache_size_kb`

Number of kilo bytes in cache.

Type: `int`. Default value: `0x2000`.

`ccn_cache.cache_state_modelled`

Model the cache state.

Type: `bool`. Default value: `0x1`.

`ccn_cache.force_on_from_start`

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'.

Type: `bool`. Default value: `0x0`.

ccn_cache.number_of_phantom_entries

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.

Type: `int`. Default value: `0x20`.

ccn_cache.periphbase

Value for PERIPHBASE. Only bits [43:24] are used.

Type: `int`. Default value: `0x2c000000`.

ccn_cache.reentrancy_support

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'.

Type: `string`. Default value: `"env"`.

ccn_cache.sbas_bridge_present

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

Type: `bool`. Default value: `0x1`.

ccn_cache.sbsx_bridge_present

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

Type: `bool`. Default value: `0x1`.

4.10.10 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 4-357: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CCN512 contains the following CADI targets:

- CCN5XX
- CCNCache

CCN512 contains the following MTI components:

- [CCNCache](#)
- [CCNRegisterSet](#)
- [PVBusExclusiveMonitor](#)

- [PVBusMapper](#)
- [PVBusSlave](#)

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`

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 `ccn_cache` subcomponent are not accessible in System Canvas.

Ports for CCN512

Table 4-358: Ports

Name	Protocol	Type	Description
<code>pvbuss_m_hni[2]</code>	PVBus	Master	HNI downstream ports.
<code>pvbuss_m_snf[4]</code>	PVBus	Master	SNF downstream ports.
<code>pvbuss_s_rnf[12]</code>	PVBus	Slave	RNF upstream ports.
<code>pvbuss_s_rni[24]</code>	PVBus	Slave	RNI upstream ports.
<code>reset_in</code>	Signal	Slave	Reset signal.

Parameters for CCNCache

`ccn_cache.acchannelen_rnf`

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.
Type: `int`. Default value: `0xf`.

`ccn_cache.acchannelen_rni`

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.
Type: `int`. Default value: `0xffff`.

`ccn_cache.cache_size_kb`

Number of kilo bytes in cache.

Type: `int`. Default value: `0x2000`.

`ccn_cache.cache_state_modelled`

Model the cache state.

Type: `bool`. Default value: `0x1`.

`ccn_cache.force_on_from_start`

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'.

Type: `bool`. Default value: `0x0`.

`ccn_cache.number_of_phantom_entries`

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.

Type: `int`. Default value: `0x20`.

`ccn_cache.periphbase`

Value for PERIPHBASE. Only bits [43:24] are used.

Type: `int`. Default value: `0x2c000000`.

`ccn_cache.reentrancy_support`

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'.

Type: `string`. Default value: `"env"`.

`ccn_cache.sbas_bridge_present`

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

Type: `bool`. Default value: `0x1`.

`ccn_cache.sbsx_bridge_present`

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

Type: `bool`. Default value: `0x1`.

4.10.11 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 4-359: IP revisions support

Revision	Quality level
r2p0	Full support
r1p0	Full support
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CI700 contains the following CADI targets:

- CI700
- CMN_TAG_CACHE

CI700 contains the following MTI components:

- [CI700](#)
- [CMN600Cache](#)
- [CMNTAGCACHECADI](#)
- [PVBusExclusiveMonitor](#)
- [PVBusMapper](#)
- [PVBusSlave](#)

Configuring and using the model

- 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 r1p6-00rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm® Developer. To download it, contact [Arm Technical Support](#).
- The model revision is read from the topology specified by the `mesh_config_file` parameter.
- 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 CI700 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.

- 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 `HNF_ABF_PR.abf_mode` is Reserved and `HNF_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
 - Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact [Arm Technical Support](#) with questions about support for this error condition.
 - The optional interrupt `INTREQPPU` is not supported.

Model limitations

- Out of scope:
 - PMU counters are not supported. Counter registers are implemented as RAZ.
 - QoS is not supported and all related registers are RAZ/WI.
 - Error injection and error generation are not supported. All error registers are RAZ/WI.
 - Power, clock, and interrupt signals are not supported.

The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch Target operations are not supported.

- 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.
- 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
```

- Only hashing granularities of 4096B and above are supported.
- HN-Fs with different SLC sizes in the same configuration are not supported.
- MPAM features are not supported in the model, and the configuration-dependent register reset values do not match the RTL. Some of the registers (`por_hnf_mpam_idr` to `por_hnf_mpam_mbwumon_idr`) shared between security modes are not present in MPAM_S.
- The MPAM reset values described in `por_hnf_mpam_ns_por_hnf_mpam_idr[31:24]` default to the not-supported state. The model parameter `hnf_mpam_idr_override` can be used to override this but does not implement any of the functionality.
- MPAM_S `secure_register_groups_override` is not supported.
- MTU `secure_register_groups_override` is not supported.
- 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.
- 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.
- 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.
- No support for RAS.

Ports for CI700

Table 4-360: Ports

Name	Protocol	Type	Description
<code>event_clusters</code>	Signal	Peer	CPU event communication signal from the clusters.
<code>pvbus_m_hni[4]</code>	PVBus	Master	HNI downstream ports.
<code>pvbus_m_snf[8]</code>	PVBus	Master	SNF downstream port.
<code>pvbus_s_apb</code>	PVBus	Slave	APB interface port.
<code>pvbus_s_rnf[8]</code>	PVBus	Slave	RNF upstream ports.
<code>pvbus_s_rni[24]</code>	PVBus	Slave	RNI upstream ports.
<code>reset_in</code>	Signal	Slave	Reset signal.
<code>rnf_sci_s[8]</code>	SystemCoherencyInterface	Slave	System Coherency port to move the RN-F in and out of the coherency domain.
<code>rnf_upstream_reset_in[8]</code>	Signal	Slave	Used by the interconnect to determine the reset state of the RNF master to check an aspect of reset sequencing. If <code>debug_force_snoop</code> is used, this port must be connected.
<code>rni_sci_s[24]</code>	SystemCoherencyInterface	Slave	System Coherency port to move the RN-D in and out of the coherency domain.
<code>rni_upstream_reset_in[24]</code>	Signal	Slave	Used by the interconnect to determine the reset state of the RNI master to check an aspect of reset sequencing. If <code>debug_force_snoop</code> is used, this port must be connected.

Parameters for CI700

`acchannelen_rnf`

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'.

Type: `string`. Default value: "0".

`acchannelen_rni`

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'.

Type: `string`. Default value: "0".

bypass_tag_cache

If true, CI700 will bypass the tag cache component which provides the MTE support.

Type: `bool`. Default value: `0x0`.

cache_state_modelled

Model the cache state.

Type: `bool`. Default value: `0x0`.

debug_force_snoop

The CI700 interconnect will normally start with snooping disabled. The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface. For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports. However, for connections that are `_not_` managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it. NOTE: This parameter is for debug purpose only Do not use this parameter instead of correctly configuring CMN. Refer CI700 documentation page for more info about this parameter in fast models reference manual.

Type: `bool`. Default value: `0x0`.

enable_logger

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: `bool`. Default value: `0x0`.

enable_rnsam_to_hnf_wider_hash

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].

Type: `bool`. Default value: `0x0`.

force_rnsam_internal

Force all RNSAMs to be internal independently of the mesh topology.

Type: `bool`. Default value: `0x1`.

hnf_mpam_idr_override

Set to override `hnf_mpam_idr[31:24]` value. Bit[28] is Reserved and is ignored.

Type: `int`. Default value: `0x0`.

mesh_config_file

Name of a file containing mesh placement of CI700 components.

Type: `string`. Default value: `""`.

periphbase

Value for PERIPHBASE. Bits [27:0] are treated 0.

Type: `int`. Default value: `0x20000000`.

print_cmn_ccix_config

Print information about the CCIX configuration.

Type: `bool`. Default value: `0x0`.

print_cmn_config

Print the mesh topology and children pointers acquired from the YAML file.

Type: `bool`. Default value: `0x0`.

revision

Component revision. Currently supports r2p0, r1p0, r0p0. The version in the `mesh_config_file` takes priority.

Type: `string`. Default value: `"r0p0"`.

rnf_sci_enable

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'.

Type: `string`. Default value: `"0x0"`.

rni_sci_enable

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'.

Type: `string`. Default value: `"0x0"`.

show_banner

Show component banner: 0=>supress entire banner, 1=>suppress config file, 2+=>show full banner.

Type: `int`. Default value: `0x2`.

skip_cmn_config_check

Skip any topology configuration checks. The maximum number of devices per type not verified.

Type: `bool`. Default value: `0x0`.

use_yaml_periphbase

Use yaml param CFGM_PERIPHBASE_PARAM to specify periphbase address. If false, model parameter 'periphbase' will be used.

Type: `bool`. Default value: `0x0`.

yaml_has_node_addresses

Does the top-level YAML file describe node-addresses ?.

Type: `bool`. Default value: `0x0`.

4.10.12 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 4-361: IP revisions support

Revision	Quality level
r1p1	Full support
r3p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CMN600 contains the following CADI targets:

- CMN600
- CMN600Cache

CMN600 contains the following MTI components:

- [CMN600](#)
- [CMN600Cache](#)
- [PVBusExclusiveMonitor](#)
- [PVBusMapper](#)
- [PVBusSlave](#)

About CMN600

For information about the changes in this release, see the Fast Models Portfolio release notes.

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](#).

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 CMN600 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: 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 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch Target operations are not supported.
- 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.
- 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 [5. Plug-ins for Fast Models](#) on page 1617 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.
- 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.

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 4-362: Ports

Name	Protocol	Type	Description
event_clusters	Signal	Peer	CPU event communication signal from the clusters.
event_downstream_link_signal[6]	Signal	Master	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual "links" on underlying PCIe hardware bus.
event_upstream_link_signal[6]	Signal	Slave	Event from the Hub towards the CMN
pvbus_m_cml	PVBus	Master	CML downstream ports
pvbus_m_cml_cfg	PVBus	Master	CML downstream hub configuration port
pvbus_m_hni[8]	PVBus	Master	HNI downstream ports.
pvbus_m_snf[16]	PVBus	Master	SNF downstream port.
pvbus_s_cml	PVBus	Slave	CML upstream ports
pvbus_s_rnf[64]	PVBus	Slave	RNF upstream ports.
pvbus_s_rni[96]	PVBus	Slave	RNI upstream ports.
reset_in	Signal	Slave	Reset signal.
rnf_sci_s[64]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in[64]	Signal	Slave	Used by the interconnect to determine the reset state of the RNF master 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 master to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

Parameters for CMN600

acchannelen_rnf

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'.

Type: string. Default value: "0".

acchannelen_rni

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'.

Type: string. Default value: "0".

cache_state_modelled

Model the cache state.

Type: bool. Default value: 0x0.

debug_force_snoop

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. NOTE: This parameter is for debug purpose only Do not use this parameter instead of correctly configuring CMN. Refer CMN600 documentation page for more info about this parameter in fast models reference manual.

Type: bool. Default value: 0x0.

disable_CML_port

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

Type: bool. Default value: 0x0.

enable_logger

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: bool. Default value: 0x0.

enable_rnsam_to_hnf_wider_hash

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].

Type: bool. Default value: 0x0.

force_rnsam_internal

Force all RNSAMs to be internal independently of the mesh topology.

Type: bool. Default value: 0x1.

mesh_config_file

Name of a file containing mesh placement of CMN600 components.

Type: string. Default value: "".

periphbase

Value for PERIPHBASE. Bits [25:0] are treated 0.

Type: int. Default value: 0x20000000.

print_cmn_ccix_config

Print information about the CCIX configuration.

Type: bool. Default value: 0x0.

print_cmn_config

Print the mesh topology and children pointers acquired from the YAML file.

Type: bool. Default value: 0x0.

revision

Component revision. Currently supports r1p1, r3p0.

Type: `string`. Default value: `"r1p1"`.

rnf_sci_enable

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'.

Type: `string`. Default value: `"0x0"`.

rni_sci_enable

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'.

Type: `string`. Default value: `"0x0"`.

show_banner

Show component banner: 0=>supress entire banner, 1=>suppress config file, 2+=>show full banner.

Type: `int`. Default value: `0x2`.

skip_cmn_config_check

Skip any topology configuration checks. The maximum number of devices per type not verified.

Type: `bool`. Default value: `0x0`.

use_yaml_periphbase

Use yaml param CFGM_PERIPHBASE_PARAM to specify periphbase address. If false, model parameter 'periphbase' will be used.

Type: `bool`. Default value: `0x0`.

4.10.13 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 4-363: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CMN600AE contains the following CADI targets:

- CMN600AE

CMN600AE contains the following MTI components:

- [CMN600AE](#)

- [CMN600Cache](#)
- [PVBusExclusiveMonitor](#)
- [PVBusMapper](#)
- [PVBusSlave](#)

About CMN600AE

For information about the changes in this release, see the Fast Models Portfolio release notes.

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](#).

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:
 - -8: OKB when `HN*_SLC_NUM_WAYS_PARAM=16`



This value is not supported by the model.

-
- -2: 128KB when `HN*_SLC_NUM_WAYS_PARAM=16`
 - -1: 256KB when `HN*_SLC_NUM_WAYS_PARAM=16`
 - 0: 512KB when `HN*_SLC_NUM_WAYS_PARAM=16`
 - 1: 1MB when `HN*_SLC_NUM_WAYS_PARAM=16`
 - 2: 2MB when `HN*_SLC_NUM_WAYS_PARAM=16`
 - 3: 3MB when `HN*_SLC_NUM_WAYS_PARAM=12`
 - 3: 4MB when `HN*_SLC_NUM_WAYS_PARAM=16`
 - Maximum number of nodes that have been verified are:
 - 7 `RN-Fs`
 - 2 `RN-Ds`

- 3 RN-Is
- 1 HN-I
- 4 HN-Fs
- 1 SN-F
- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
 - In the Fast Model, any HN-F in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the HN-F belongs to.
 - Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.
 - When `HNF_ABF_PR.abf_mode` is Reserved and `HNF_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
 - Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact [Arm Technical Support](#) with questions about support for this error condition.
 - The optional interrupt `INTREQPPU` is not supported.
- This model implements the following AE-specific features:
 - Memory Protection Unit (MPU), with limitations listed in the Model limitations section.
 - FuSa error logging and reporting using a Fault Management Unit (FMU) and Fault Detection and Control (FDC).
 - Dedicated APB interface into FMU for fault diagnostics and control.

Model limitations

- Out of scope:
 - PMU counters are not supported. Counter registers are implemented as RAZ.
 - QoS is not supported and all related registers are RAZ/WI.
 - Error injection and error generation are not supported. All error registers are RAZ/WI.
 - Power, clock, and interrupt signals are not supported.

The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch Target operations are not supported.
- 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.
- 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 - 0x0000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example HNFSAM_Mem_Map:

```
0x00000001000 - 0x0000002000 SNF0
0x0000002000 - 0x0000003000 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 [5. Plug-ins for Fast Models](#) on page 1617 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.
- 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 4-364: 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 master 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 master to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

Parameters for CMN600AE

acchannelen_rnf

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'.

Type: `string`. Default value: "0".

acchannelen_rni

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'.

Type: `string`. Default value: "0".

cache_state_modelled

Model the cache state.

Type: `bool`. Default value: 0x0.

debug_force_snoop

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. NOTE: This parameter is for debug purpose only Do not use this parameter instead of correctly configuring CMN. Refer CMN600AE documentation page for more info about this parameter in fast models reference manual.

Type: `bool`. Default value: 0x0.

disable_CML_port

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

Type: `bool`. Default value: 0x0.

enable_logger

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: `bool`. Default value: 0x0.

enable_rnsam_to_hnf_wider_hash

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].

Type: `bool`. Default value: 0x0.

fdc_key

Por_fdc_key register value is checked against this key.

Type: `int`. Default value: 0x0.

fm_u_key

Por_fm_u_key register value is checked against this key.

Type: `int`. Default value: 0x0.

force_rnsam_internal

Force all RNSAMs to be internal independently of the mesh topology.

Type: `bool`. Default value: `0x1`.

mesh_config_file

Name of a file containing mesh placement of CMN600AE components.

Type: `string`. Default value: `""`.

number_of_mpu_programmable_regions

Number of MPU programmable regions. Valid values are 0, 8, 16 and 32.

Type: `int`. Default value: `0x20`.

periphbase

Value for PERIPHBASE. Bits [25:0] are treated 0.

Type: `int`. Default value: `0x20000000`.

print_cm_n_ccix_config

Print information about the CCIX configuration.

Type: `bool`. Default value: `0x0`.

print_cm_n_config

Print the mesh topology and children pointers acquired from the YAML file.

Type: `bool`. Default value: `0x0`.

revision

Component revision. Currently supports r1p0.

Type: `string`. Default value: `"r1p0"`.

rnf_sci_enable

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'.

Type: `string`. Default value: `"0x0"`.

rni_sci_enable

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'.

Type: `string`. Default value: `"0x0"`.

show_banner

Show component banner: 0=>supress entire banner, 1=>suppress config file, 2+=>show full banner.

Type: `int`. Default value: `0x2`.

skip_cm_n_config_check

Skip any topology configuration checks. The maximum number of devices per type not verified.

Type: `bool`. Default value: `0x0`.

use_yaml_periphbase

Use yaml param CFGM_PERIPHBASE_PARAM to specify periphbase address. If false, model parameter 'periphbase' will be used.

Type: `bool`. Default value: `0x0`.

4.10.14 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 4-365: IP revisions support

Revision	Quality level
N/A	Preliminary support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CMN600CMLHub contains the following CADI targets:

- CMN600CMLHub

CMN600CMLHub contains the following MTI components:

- [CMN600CML](#)
- [PVBusExclusiveMonitor](#)
- [PVBusMapper](#)
- [PVBusSlave](#)
- [PVCache](#)

Ports for CMN600CMLHub

Table 4-366: 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.
pvbus_m[4]	PVBus	Master	Downstream CCIX port.
pvbus_s[4]	PVBus	Slave	Upstream CCIX ports.
pvbus_s_cfg[4]	PVBus	Slave	Upstream config ports.
reset_signal	Signal	Slave	Reset signal.

Parameters for CMN600CMLHub

cache_state_modelled

Model the cache state.

Type: `bool`. Default value: `0x0`.

enable_logger

Enable PVBusLoggers for the downstream ports in the CMN model.

Type: `bool`. Default value: `0x0`.

4.10.15 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 4-367: IP revisions support

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CMN650 contains the following CADI targets:

- CMN650

CMN650 contains the following MTI components:

- [CMN600Cache](#)
- [CMN650](#)
- [PVBusExclusiveMonitor](#)
- [PVBusMapper](#)
- [PVBusSlave](#)

About CMN650

- For information about the changes in this release, see the Fast Models Portfolio release notes.
- 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](#).

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`.



This value is not supported by the model.

- -2: 128KB where `HN*_SLC_NUM_WAYS_PARAM=16`
- -1: 256KB where `HN*_SLC_NUM_WAYS_PARAM=16`
- 0: 512KB where `HN*_SLC_NUM_WAYS_PARAM=16`
- 1: 1MB where `HN*_SLC_NUM_WAYS_PARAM=16`
- 2: 2MB where `HN*_SLC_NUM_WAYS_PARAM=16`
- 3: 3MB where `HN*_SLC_NUM_WAYS_PARAM=12`
- 3: 4MB where `HN*_SLC_NUM_WAYS_PARAM=16`
- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
 - In the Fast Model, any HN-F in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the HN-F belongs to.
 - Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.
 - When `HN*_ABF_PR.abf_mode` is Reserved and `HN*_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
 - Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact [Arm Technical Support](#) with questions about support for this error condition.
 - The optional interrupt `INTREQPPU` is not supported.

Model limitations

- Out of scope:
 - PMU counters are not supported. Counter registers are implemented as RAZ.
 - QoS is not supported and all related registers are RAZ/WI.
 - Error injection and error generation are not supported. All error registers are RAZ/WI.
 - Power, clock, and interrupt signals are not supported.

The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.

- Prefetch Target operations are not supported.
- 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.
- The fields in the `mxp_device_port_connect_info` register corresponding to the number of device credit slices and the number of CAL credit slices are not supported.
- SN-F hashing in HNSAM is not supported. HNSAM SN-F hashing is a single memory range mapped to multiple SN-Fs. Example HNFSAM_Mem_Map:

```
0x00000001000 - 0x00000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example HNFSAM_Mem_Map:

```
0x00000001000 - 0x00000002000 SNF0
0x00000002000 - 0x00000003000 SNF1
```

- HN-Fs with different SLC sizes in the same configuration are not supported.
- Transactions to addresses unused by Device registers in the Configuration Register Space are always routed to the Configuration Space even when the SAM is configured to use those addresses.
- GIC communication over A4S ports is not supported.
- 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 [5. Plug-ins for Fast Models](#) on page 1617 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 HND irrespective of what is programmed in the `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` fields.
- Address bit masking is not supported in RNSAM or HN-F SAM.
- For an 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 4-368: Ports

Name	Protocol	Type	Description
<code>event_clusters</code>	Signal	Peer	CPU event communication signal from the clusters.
<code>event_downstream_link_signal[10]</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[10]</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[16]</code>	PVBus	Master	HNI downstream ports.
<code>pvbust_m_snf[40]</code>	PVBus	Master	SNF downstream port.
<code>pvbust_s_apb</code>	PVBus	Slave	APB interface port.
<code>pvbust_s_cml</code>	PVBus	Slave	CML upstream ports
<code>pvbust_s_rnf[64]</code>	PVBus	Slave	RNF upstream ports.

Name	Protocol	Type	Description
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 master 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 master to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

Parameters for CMN650

acchannelen_rnf

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'.

Type: `string`. Default value: "0".

acchannelen_rni

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'.

Type: `string`. Default value: "0".

cache_state_modelled

Model the cache state.

Type: `bool`. Default value: 0x0.

debug_force_snoop

The CMN650 interconnect will normally start with snooping disabled. The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface. For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports. However, for connections that are `_not_` managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it. NOTE: This parameter is for debug purpose only Do not use this parameter instead of correctly configuring CMN. Refer CMN650 documentation page for more info about this parameter in fast models reference manual.

Type: `bool`. Default value: 0x0.

disable_CML_port

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

Type: `bool`. Default value: 0x0.

enable_logger

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: `bool`. Default value: `0x0`.

enable_rnsam_to_hnf_wider_hash

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].

Type: `bool`. Default value: `0x0`.

force_rnsam_internal

Force all RNSAMs to be internal independently of the mesh topology.

Type: `bool`. Default value: `0x1`.

hnf_mpam_idr_override

Set to override hnf_mpam_idr[31:24] value. Bit[28] is Reserved and is ignored.

Type: `int`. Default value: `0x0`.

mesh_config_file

Name of a file containing mesh placement of CMN650 components.

Type: `string`. Default value: `""`.

periphbase

Value for PERIPHBASE. Bits [27:0] are treated 0.

Type: `int`. Default value: `0x20000000`.

print_cmn_ccix_config

Print information about the CCIX configuration.

Type: `bool`. Default value: `0x0`.

print_cmn_config

Print the mesh topology and children pointers acquired from the YML file.

Type: `bool`. Default value: `0x0`.

revision

Component revision. Currently supports r1p1.

Type: `string`. Default value: `"r1p1"`.

rnf_sci_enable

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'.

Type: `string`. Default value: `"0x0"`.

rni_sci_enable

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'.

Type: `string`. Default value: `"0x0"`.

show_banner

Show component banner: 0=>supress entire banner, 1=>suppress config file, 2+=>show full banner.

Type: `int`. Default value: `0x2`.

skip_cmn_config_check

Skip any topology configuration checks. The maximum number of devices per type not verified.

Type: `bool`. Default value: `0x0`.

use_yaml_periphbase

Use yaml param CFGM_PERIPHBASE_PARAM to specify periphbase address. If false, model parameter 'periphbase' will be used.

Type: `bool`. Default value: `0x0`.

4.10.16 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 4-369: IP revisions support

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CMN650R2 contains the following CADI targets:

- CMN650R2

CMN650R2 contains the following MTI components:

- [CMN600Cache](#)
- [CMN650R2](#)
- [PVBusExclusiveMonitor](#)
- [PVBusMapper](#)
- [PVBusSlave](#)

About CMN650R2

- For information about the changes in this release, see the Fast Models Portfolio release notes.
- To configure the model, you must have installed Arm® Socrates™. The `mesh_config_file` parameter defines the mesh placement of the CHI nodes. Set it to the name of the yaml configuration file emitted by Socrates. You must use version r1p4-01rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm® Developer. To download it, contact [Arm Technical Support](#).

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: 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 `HNF_ABF_PR.abf_mode` is Reserved and `HNF_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
 - Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact [Arm Technical Support](#) with questions about support for this error condition.
 - The optional interrupt `INTREQPPU` is not supported.

Model limitations

- Out of scope:
 - PMU counters are not supported. Counter registers are implemented as RAZ.
 - QoS is not supported and all related registers are RAZ/WI.
 - Error injection and error generation are not supported. All error registers are RAZ/WI.
 - Power, clock, and interrupt signals are not supported.

The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is `5'b00000`. So the initial power state of the HN-F nodes is `NOSFSLC/OFF`. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch Target operations are not supported.
- 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.
- The fields in the `mxp_device_port_connect_info` register corresponding to the number of device credit slices and the number of CAL credit slices are not supported.
- SN-F hashing in HNSAM is not supported. HNSAM SN-F hashing is a single memory range mapped to multiple SN-Fs. Example `HNFSAM_Mem_Map`:

```
0x00000001000 - 0x0000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example `HNFSAM_Mem_Map`:

```
0x00000001000 - 0x0000002000 SNF0
0x0000002000 - 0x0000003000 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 [5. Plug-ins for Fast Models](#) on page 1617 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 HND 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.
- 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_exc1` 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 4-370: Ports

Name	Protocol	Type	Description
event_clusters	Signal	Peer	CPU event communication signal from the clusters.
event_downstream_link_signal[10]	Signal	Master	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual "links" on underlying PCIe hardware bus.
event_upstream_link_signal[10]	Signal	Slave	Event from the Hub towards the CMN
pvbus_m_cml	PVBus	Master	CML downstream ports
pvbus_m_cml_cfg	PVBus	Master	CML downstream hub configuration port
pvbus_m_hni[16]	PVBus	Master	HNI downstream ports.
pvbus_m_snf[40]	PVBus	Master	SNF downstream port.
pvbus_s_apb	PVBus	Slave	APB interface port.
pvbus_s_cml	PVBus	Slave	CML upstream ports
pvbus_s_rnf[256]	PVBus	Slave	RNF upstream ports.
pvbus_s_rni[108]	PVBus	Slave	RNI upstream ports.
reset_in	Signal	Slave	Reset signal.
rnf_sci_s[256]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in[256]	Signal	Slave	Used by the interconnect to determine the reset state of the RNF master 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 master to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

Parameters for CMN650R2

acchannelen_rnf

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'.

Type: string. Default value: "0".

acchannelen_rni

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'.

Type: string. Default value: "0".

cache_state_modelled

Model the cache state.

Type: bool. Default value: 0x0.

debug_force_snoop

The CMN650R2 interconnect will normally start with snooping disabled. The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface. For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports. However, for connections that are `_not_` managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it. NOTE: This parameter is for debug purpose only Do not use this parameter instead of correctly configuring CMN. Refer CMN650R2 documentation page for more info about this parameter in fast models reference manual.

Type: `boo1`. Default value: `0x0`.

disable_CML_port

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

Type: `boo1`. Default value: `0x0`.

enable_logger

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: `boo1`. Default value: `0x0`.

enable_rnsam_to_hnf_wider_hash

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].

Type: `boo1`. Default value: `0x0`.

force_rnsam_internal

Force all RNSAMs to be internal independently of the mesh topology.

Type: `boo1`. Default value: `0x1`.

hnf_mpam_idr_override

Set to override `hnf_mpam_idr[31:24]` value. Bit[28] is Reserved and is ignored.

Type: `int`. Default value: `0x0`.

mesh_config_file

Name of a file containing mesh placement of CMN650R2 components.

Type: `string`. Default value: `""`.

periphbase

Value for PERIPHBASE. Bits [27:0] are treated 0.

Type: `int`. Default value: `0x20000000`.

print_cmn_ccix_config

Print information about the CCIX configuration.

Type: `boo1`. Default value: `0x0`.

print_cmn_config

Print the mesh topology and children pointers acquired from the YML file.

Type: `boo1`. Default value: `0x0`.

revision

Component revision. Currently supports r2p0.

Type: `string`. Default value: `"r2p0"`.

rnf_sci_enable

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'.

Type: `string`. Default value: `"0x0"`.

rni_sci_enable

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'.

Type: `string`. Default value: `"0x0"`.

show_banner

Show component banner: 0=>supress entire banner, 1=>suppress config file, 2+=>show full banner.

Type: `int`. Default value: `0x2`.

skip_cmn_config_check

Skip any topology configuration checks. The maximum number of devices per type not verified.

Type: `bool`. Default value: `0x0`.

use_yaml_periphbase

Use yaml param CFGM_PERIPHBASE_PARAM to specify periphbase address. If false, model parameter 'periphbase' will be used.

Type: `bool`. Default value: `0x0`.

4.10.17 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 4-371: IP revisions support

Revision	Quality level
r0p0	Full support
r1p0	Preliminary support
r2p0	Preliminary support
r3p0	Preliminary support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

CMN700 contains the following CADI targets:

- CMN700

- CMN_TAG_CACHE

CMN700 contains the following MTI components:

- [CMN600Cache](#)
- [CMN700](#)
- [CMNTAGCACHECADI](#)
- [PVBusExclusiveMonitor](#)
- [PVBusMapper](#)
- [PVBusSlave](#)

Changes in 11.24.4

Parameters removed:

- `ipxact_config_file`

About the model

- For information about the changes in this release, see the Fast Models Portfolio release notes.
- 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](#).
- 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_F_ABF_PR.abf_mode` is Reserved and `HN_F_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
- Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact [Arm Technical Support](#) with questions about support for this error condition.
- The optional interrupt `INTREQFPU` is not supported.
- There is limited support for RAS:
 - Error logging and reporting functionality for HN-I, SBSX, XP, and MTU are supported.
 - RAS-related interrupts (`INTREQERRS`, `INTREQFAULTS`, `INTREQFAULTNS`, `INTREQERRNS`) have been added.
 - Central RAS interrupt-handling functionality of HN-D is supported.



Enabling RAS can impact performance. To avoid this, RAS is disabled by default in the CMN700 model. To enable RAS, set the `enable_ras` parameter to true.

- There is support for A4S Multichip routing, with limitations:
 - When the `enable_a4s` parameter is false, top-level model ports are terminated with abort handlers.
 - Set `enable_a4s` parameter to true to opt into the feature.
 - The model routes A4S transactions from the `ic_dr_a4s` port to `tx_cxs_a4s` ports according to A4S LDIDs.
 - There is 1 A4S tx/rx port for each CCG up to a maximum of 32 CCGs.
 - When the remote transaction arrives at the receiving remote CMN on `rx_cxs_a4s` ports, it is routed to the GIC A4S port (`ic_rd_a4s`).
 - The A4S LDID for the CCGs can be found by reading the `ccg_RA.unit_info` register or through model parameter `print_cmn_config`.

A4S support limitations are listed in the Model Limitations section.

The following table describes the level of support in the CMN700 model for different revision-specific features of the IP:

Table 4-372: Model support for revision-specific features

Revision	Feature	Model support
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

Revision	Feature	Model support
	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

Model limitations

- Out of scope:
 - PMU counters are not supported. Counter registers are implemented as RAZ.
 - QoS is not supported and all related registers are RAZ/WI.
 - Error injection and error generation are not supported. All error registers are RAZ/WI.
 - Power, clock, and interrupt signals are not supported, but RAS-related interrupts are supported.

The P-channel (power) signals PREQ_LOGIC, PSTATE_LOGIC, PACCEPT_LOGIC, PDENY_LOGIC, and PACTIVE_LOGIC, are not implemented. The model behaves as if PSTATE_LOGIC is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch target operations are not supported.
- 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.
- The fields in the `mxp_device_port_connect_info` register corresponding to the number of device credit slices and the number of CAL credit slices are not supported.
- SN-F hashing in HNSAM is not supported. HNSAM SN-F hashing is a single memory range mapped to multiple SN-Fs. Example HNFSAM_Mem_Map:

```
0x00000001000 - 0x00000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example HNFSAM_Mem_Map:

```
0x00000001000 - 0x00000002000 SNF0
0x00000002000 - 0x00000003000 SNF1
```

- HN-Fs with different SLC sizes in the same configuration are not supported.
- 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_iidr` 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 [5. Plug-ins for Fast Models](#) on page 1617 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 HND 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:
 - Hybrid CAL flavors are not supported.
 - 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_striping` 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.
- When both `POR_CCLA_ULL_CTL.ull_to_ull_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`.

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 4-373: 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.

Name	Protocol	Type	Description
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 master 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 master 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

acchannelen_rnf

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'.

Type: `string`. Default value: "0".

acchannelen_rni

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'.

Type: `string`. Default value: "0".

cache_state_modelled

Model the cache state.

Type: `bool`. Default value: 0x0.

debug_force_snoop

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. NOTE: This parameter is for debug purpose only Do not use this parameter instead of correctly configuring CMN. Refer CMN700 documentation page for more info about this parameter in fast models reference manual.

Type: `bool`. Default value: 0x0.

disable_CML_port

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

Type: `bool`. Default value: 0x0.

enable_a4s

Enables A4S ports for GIC multi-chip routing.

Type: bool. Default value: 0x0.

enable_logger

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: bool. Default value: 0x0.

enable_ras

Enables RAS. There is an impact on performance when RAS is enabled.

Type: bool. Default value: 0x0.

enable_rnsam_to_hnf_wider_hash

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].

Type: bool. Default value: 0x0.

force_rnsam_internal

Force all RNSAMs to be internal independently of the mesh topology.

Type: bool. Default value: 0x1.

hnf_mpam_idr_override

Set to override hnf_mpam_idr[31:24] value. Bit[28] is Reserved and is ignored.

Type: int. Default value: 0x0.

ipxact_config_file

null.

Type: string. Default value: "".

mesh_config_file

Name of a file containing mesh placement of CMN700 components.

Type: string. Default value: "".

periphbase

Value for PERIPHBASE. Bits [27:0] are treated 0.

Type: int. Default value: 0x20000000.

print_cmn_ccix_config

Print information about the CCIX configuration.

Type: bool. Default value: 0x0.

print_cmn_config

Print the mesh topology and children pointers acquired from the YML file.

Type: bool. Default value: 0x0.

register_traces_for_ccg_apb_accesses

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).

Type: bool. Default value: 0x0.

revision

Component revision. Currently supports r0p0, r1p0, r2p0, r3p0. The version in the mesh_config_file takes priority.

Type: `string`. Default value: "r0p0".

rnf_sci_enable

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'.

Type: `string`. Default value: "0x0".

rni_sci_enable

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'.

Type: `string`. Default value: "0x0".

show_banner

Show component banner: 0=>supress entire banner, 1=>suppress config file, 2+=>show full banner.

Type: `int`. Default value: 0x2.

skip_cmn_config_check

Skip any topology configuration checks. The maximum number of devices per type not verified.

Type: `bool`. Default value: 0x0.

use_yaml_periphbase

Use yaml param CFGM_PERIPHBASE_PARAM to specify periphbase address. If false, model parameter 'periphbase' will be used.

Type: `bool`. Default value: 0x0.

yaml_has_node_addresses

Does the top-level YAML file describe node-addresses ?.

Type: `bool`. Default value: 0x0.

4.10.18 DMA350

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-374: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

DMA350 contains the following CADI targets:

- DMA350

DMA350 contains the following MTI components:

- [DMA350](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

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 4-375: 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

Name	Protocol	Type	Description
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
irq_comb_nonsec	Signal	Master	Nonsecure IRQ Signal
irq_comb_nonsec_err	Signal	Master	Nonsecure error IRQ Signal
irq_comb_sec	Signal	Master	Secure IRQ Signal
irq_comb_sec_err	Signal	Master	Secure error IRQ Signal
irq_sec_viol_err	Signal	Master	Security violation IRQ Signal
pvbus_m0	PVBus	Master	AXI5 Master 0 Interface
pvbus_m1	PVBus	Master	AXI5 Master 1 Interface
pvbus_s	PVBus	Slave	APB4 Slave Interface
pvbus_stream_in[16]	PVBus	Slave	AXI-Stream In Interface
pvbus_stream_out[16]	PVBus	Master	AXI-Stream Out Interface
reset_in	Signal	Slave	Ada DMA asynchronous reset
trig_in_ack[32]	Signal	Master	Trigger In Acknowledgement Interface
trig_in_ack_type[32]	Value	Master	-
trig_in_req[32]	Signal	Slave	Trigger In Request Interface
trig_in_req_type[32]	Value	Slave	-
trig_out_ack[32]	Signal	Slave	Trigger Out Acknowledgement Interface
trig_out_req[32]	Signal	Master	Trigger Out Request Interface

4.10.19 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 4-376: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

DMC500 contains the following CADI targets:

- DMC500

DMC500 contains the following MTI components:

- [DMC-500](#)
- [PVBusSlave](#)

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 4-377: 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.

Name	Protocol	Type	Description
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.

4.10.20 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 4-378: IP revisions support

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

DMC520 contains the following CADI targets:

- DMC520

DMC520 contains the following MTI components:

- [DMC-520](#)
- [PVBusSlave](#)

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 4-379: 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.
<code>scrub_event_in[8]</code>	Signal	Slave	Scrub event n trigger.
<code>scrub_event_out[8]</code>	Signal	Master	Scrub event n triggered.

4.10.21 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 4-380: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

DMC620 contains the following CADI targets:

- DMC620

DMC620 contains the following MTI components:

- [DMC-620](#)
- [PVBusSlave](#)

About DMC620

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 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 4-381: 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-620.
<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>interrupt_cfh_master</code>	Signal	Master	The DMC has detected and corrected a single bit error on the RAM access.
<code>interrupt_combined_oflow_master</code>	Signal	Master	The DMC has detected a counter overflow.
<code>interrupt_fh_master</code>	Signal	Master	The DMC has detected a double bit error on the RAM access.
<code>reset_signal</code>	Signal	Slave	DMC reset.

4.10.22 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 4-382: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

DMC_400 contains the following CADI targets:

- DMC_400

DMC_400 contains the following MTI components:

- [PVBusMapper](#)
- [PVBusSlave](#)

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 4-383: 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

diagnostics

Diagnostics.

Type: int. Default value: 0x0.

ECC_SUPPORT

Does the controller support ECC?.

Type: bool. Default value: 0x1.

IF_CHIP0

Set this parameter to 0 if memory is connected.
Type: `int`. Default value: `-0x1`.

IF_CHIP1

Set this parameter to 0 if memory is connected.
Type: `int`. Default value: `-0x1`.

IF_CHIP2

Set this parameter to 0 if memory is connected.
Type: `int`. Default value: `-0x1`.

IF_CHIP3

Set this parameter to 0 if memory is connected.
Type: `int`. Default value: `-0x1`.

MEMORY_WIDTH

Valid widths are 16, 32 or 64 bits.
Type: `int`. Default value: `0x20`.

revision_string

Revision.
Type: `string`. Default value: `"r0p1"`.

4.10.23 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 4-384: IP revisions support

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

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 `resetsn_in` signal is active-LOW.

Ports for EthosU55**Table 4-385: Ports**

Name	Protocol	Type	Description
<code>clk_in</code>	ClockSignal	Slave	NPU clock signal
<code>irq_out</code>	Signal	Master	Sends interrupt requests to the external host application processor
<code>popl_in</code>	Signal	Slave	The Power-On-Reset Privilege Level (PORPL). LOW means User level. HIGH means Privileged level
<code>posl_in</code>	Signal	Slave	The Power-On-Reset Security Level (PORSL). LOW means Secure. HIGH means Non-secure.
<code>pvbuss_m0</code>	PVBus	Master	Port 0 for NPU to access external memory
<code>pvbuss_m1</code>	PVBus	Master	Port 1 for NPU to access external memory
<code>pvbuss_s</code>	PVBus	Slave	Port to access NPU control registers
<code>resetsn_in</code>	Signal	Slave	NPU reset signal

4.10.24 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 4-386: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

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.



- The EthosU65 model does not expose its registers through CADI or Iris.
- The `resetsn_in` signal is active-LOW.

Ports for EthosU65**Table 4-387: 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.
pvbust_m0	PVBus	Master	Port 0 for NPU to access external memory
pvbust_m1	PVBus	Master	Port 1 for NPU to access external memory
pvbust_s	PVBus	Slave	Port to access NPU control registers
resetsn_in	Signal	Slave	NPU reset signal

4.10.25 Firewall

Firewall IP. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-388: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Firewall contains the following CADI targets:

- Firewall

Firewall contains the following MTI components:

- [Firewall](#)
- [PVBusLogger](#)
- [PVBusMapper](#)
- [PVBusSlave](#)

Ports for Firewall

Table 4-389: 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	-

4.10.26 GIC500

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 4-390: IP revisions support

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC500 contains the following CADI targets:

- GIC500

GIC500 contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3InterruptTranslationService](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)

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 Arm®v8-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.

You must configure some parameters in order to use the GIC500 component. For example:

```
gic500: GIC500(
    "num_clusters" = 2,
    "cpus_per_cluster_0" = 4,
    "cpus_per_cluster_1" = 4,
    "reg-base" = 0x2c200000,
    "SPI-count" = 256
);
```



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 the [MSIRewriter](#) component.

MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

Table 4-391: `pvbus_s` port

PVBus attribute	Bits used	Property encoded	Notes
MasterID	Bits[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 4-392: Ports

Name	Protocol	Type	Description
<code>cfgsdisable</code>	Signal	Slave	Disable some SPIs signal.
<code>cpu_active_0[8]</code>	Signal	Slave	<code>cpu_active</code> pins of cluster 0.
<code>cpu_active_1[8]</code>	Signal	Slave	<code>cpu_active</code> pins of cluster 1.
<code>cpu_active_10[8]</code>	Signal	Slave	<code>cpu_active</code> pins of cluster 10.
<code>cpu_active_11[8]</code>	Signal	Slave	<code>cpu_active</code> pins of cluster 11.
<code>cpu_active_12[8]</code>	Signal	Slave	<code>cpu_active</code> pins of cluster 12.
<code>cpu_active_13[8]</code>	Signal	Slave	<code>cpu_active</code> pins of cluster 13.
<code>cpu_active_14[8]</code>	Signal	Slave	<code>cpu_active</code> pins of cluster 14.
<code>cpu_active_15[8]</code>	Signal	Slave	<code>cpu_active</code> pins of cluster 15.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Reset.
spi_in[988]	Signal	Slave	Shared peripheral interrupts.
wake_request_0[8]	Signal	Master	Power management outputs of cluster 0.
wake_request_1[8]	Signal	Master	Power management outputs of cluster 1.
wake_request_10[8]	Signal	Master	Power management outputs of cluster 10.
wake_request_11[8]	Signal	Master	Power management outputs of cluster 11.
wake_request_12[8]	Signal	Master	Power management outputs of cluster 12.
wake_request_13[8]	Signal	Master	Power management outputs of cluster 13.
wake_request_14[8]	Signal	Master	Power management outputs of cluster 14.
wake_request_15[8]	Signal	Master	Power management outputs of cluster 15.
wake_request_16[8]	Signal	Master	Power management outputs of cluster 16.

Name	Protocol	Type	Description
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

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: bool. Default value: 0x0.

common-vPE-table-affinity

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.

Type: string. Default value: "".

cpus_per_cluster_0

Number of cores within cluster 0.

Type: int. Default value: 0x1.

cpus_per_cluster_1

Number of cores within cluster 1.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_10`

Number of cores within cluster 10.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_11`

Number of cores within cluster 11.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_12`

Number of cores within cluster 12.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_13`

Number of cores within cluster 13.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_14`

Number of cores within cluster 14.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_15`

Number of cores within cluster 15.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_16`

Number of cores within cluster 16.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_17`

Number of cores within cluster 17.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_18`

Number of cores within cluster 18.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_19`

Number of cores within cluster 19.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_2`

Number of cores within cluster 2.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_20`

Number of cores within cluster 20.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_21`

Number of cores within cluster 21.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_22

Number of cores within cluster 22.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_23

Number of cores within cluster 23.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_24

Number of cores within cluster 24.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_25

Number of cores within cluster 25.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_26

Number of cores within cluster 26.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_27

Number of cores within cluster 27.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_28

Number of cores within cluster 28.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_29

Number of cores within cluster 29.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_3

Number of cores within cluster 3.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_30

Number of cores within cluster 30.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_31

Number of cores within cluster 31.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_4

Number of cores within cluster 4.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_5

Number of cores within cluster 5.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_6

Number of cores within cluster 6.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_7

Number of cores within cluster 7.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_8

Number of cores within cluster 8.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_9

Number of cores within cluster 9.

Type: `int`. Default value: `0x1`.

delay-ITS-accesses

Delay accesses from the ITS until GICR_SYNCRR is read.

Type: `bool`. Default value: `0x1`.

delay-redistributor-accesses

Delay memory accesses from the redistributor until GICR_SYNCRR is read.

Type: `bool`. Default value: `0x1`.

enable_protocol_checking

Enable/disable protocol checking at cpu interface.

Type: `bool`. Default value: `0x0`.

enabled

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

GICD_ITARGETSR-RAZWI

If true, the GICD_ITARGETSR registers are RAZ/WI.

Type: `bool`. Default value: `0x0`.

has-two-security-states

If true, has two security states.

Type: `bool`. Default value: `0x1`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `int`. Default value: `0x0`.

ITS-threaded-command-queue

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

num_clusters

Number of implemented affinity level1 clusters.

Type: `int`. Default value: `0x1`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

redistributor-threaded-sync

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

reg-base

GIC500 base address.

Type: `int`. Default value: `0x2c010000`.

SPI-count

Number of SPIs that are implemented.

Type: `int`. Default value: `0xe0`.

using-generated-memorymap

Use the generated memorymap for the GIC500 and warn if superfluous parameters are passed.

Type: `bool`. Default value: `0x1`.

wakeup-on-reset

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.

Type: `bool`. Default value: `0x0`.

4.10.27 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 4-393: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC500_ClusterPorts contains the following CADI targets:

- GIC500

GIC500_ClusterPorts contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3InterruptTranslationService](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)

Ports for GIC500_ClusterPorts

Table 4-394: 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.

Name	Protocol	Type	Description
cpu_active_4[8]	Signal	Slave	cpu_active pins of cluster 4.
cpu_active_5[8]	Signal	Slave	cpu_active pins of cluster 5.
cpu_active_6[8]	Signal	Slave	cpu_active pins of cluster 6.
cpu_active_7[8]	Signal	Slave	cpu_active pins of cluster 7.
cpu_active_8[8]	Signal	Slave	cpu_active pins of cluster 8.
cpu_active_9[8]	Signal	Slave	cpu_active pins of cluster 9.
po_reset	Signal	Slave	Power on reset.
ppi16_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 0.
ppi16_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 1.
ppi16_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 10.
ppi16_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 11.
ppi16_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 12.
ppi16_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 13.
ppi16_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 14.
ppi16_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 15.
ppi16_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 16.
ppi16_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 17.
ppi16_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 18.
ppi16_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 19.
ppi16_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 2.
ppi16_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 20.
ppi16_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 21.
ppi16_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 22.
ppi16_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 23.
ppi16_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 24.
ppi16_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 25.
ppi16_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 26.
ppi16_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 27.
ppi16_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 28.
ppi16_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 29.
ppi16_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 3.
ppi16_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 30.
ppi16_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 31.
ppi16_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 4.
ppi16_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 5.
ppi16_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 6.
ppi16_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 7.
ppi16_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 8.
ppi16_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 9.

Name	Protocol	Type	Description
ppi17_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 0.
ppi17_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 1.
ppi17_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 10.
ppi17_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 11.
ppi17_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 12.
ppi17_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 13.
ppi17_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 14.
ppi17_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 15.
ppi17_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 16.
ppi17_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 17.
ppi17_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 18.
ppi17_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 19.
ppi17_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 2.
ppi17_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 20.
ppi17_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 21.
ppi17_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 22.
ppi17_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 23.
ppi17_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 24.
ppi17_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 25.
ppi17_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 26.
ppi17_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 27.
ppi17_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 28.
ppi17_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 29.
ppi17_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 3.
ppi17_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 30.
ppi17_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 31.
ppi17_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 4.
ppi17_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 5.
ppi17_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 6.
ppi17_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 7.
ppi17_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 8.
ppi17_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 9.
ppi18_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 0.
ppi18_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 1.
ppi18_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 10.
ppi18_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 11.
ppi18_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 12.
ppi18_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 13.
ppi18_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 14.

Name	Protocol	Type	Description
ppi18_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 15.
ppi18_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 16.
ppi18_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 17.
ppi18_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 18.
ppi18_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 19.
ppi18_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 2.
ppi18_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 20.
ppi18_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 21.
ppi18_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 22.
ppi18_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 23.
ppi18_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 24.
ppi18_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 25.
ppi18_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 26.
ppi18_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 27.
ppi18_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 28.
ppi18_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 29.
ppi18_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 3.
ppi18_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 30.
ppi18_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 31.
ppi18_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 4.
ppi18_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 5.
ppi18_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 6.
ppi18_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 7.
ppi18_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 8.
ppi18_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 9.
ppi19_in_0[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 0.
ppi19_in_1[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 1.
ppi19_in_10[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 10.
ppi19_in_11[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 11.
ppi19_in_12[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 12.
ppi19_in_13[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 13.
ppi19_in_14[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 14.
ppi19_in_15[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 15.
ppi19_in_16[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 16.
ppi19_in_17[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 17.
ppi19_in_18[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 18.
ppi19_in_19[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 19.
ppi19_in_2[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 2.
ppi19_in_20[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 20.

Name	Protocol	Type	Description
ppi19_in_21[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 21.
ppi19_in_22[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 22.
ppi19_in_23[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 23.
ppi19_in_24[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 24.
ppi19_in_25[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 25.
ppi19_in_26[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 26.
ppi19_in_27[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 27.
ppi19_in_28[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 28.
ppi19_in_29[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 29.
ppi19_in_3[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 3.
ppi19_in_30[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 30.
ppi19_in_31[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 31.
ppi19_in_4[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 4.
ppi19_in_5[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 5.
ppi19_in_6[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 6.
ppi19_in_7[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 7.
ppi19_in_8[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 8.
ppi19_in_9[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 9.
ppi20_in_0[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 0.
ppi20_in_1[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 1.
ppi20_in_10[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 10.
ppi20_in_11[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 11.
ppi20_in_12[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 12.
ppi20_in_13[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 13.
ppi20_in_14[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 14.
ppi20_in_15[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 15.
ppi20_in_16[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 16.
ppi20_in_17[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 17.
ppi20_in_18[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 18.
ppi20_in_19[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 19.
ppi20_in_2[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 2.
ppi20_in_20[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 20.
ppi20_in_21[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 21.
ppi20_in_22[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 22.
ppi20_in_23[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 23.
ppi20_in_24[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 24.
ppi20_in_25[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 25.
ppi20_in_26[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 26.
ppi20_in_27[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 27.

Name	Protocol	Type	Description
ppi20_in_28[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 28.
ppi20_in_29[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 29.
ppi20_in_3[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 3.
ppi20_in_30[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 30.
ppi20_in_31[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 31.
ppi20_in_4[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 4.
ppi20_in_5[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 5.
ppi20_in_6[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 6.
ppi20_in_7[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 7.
ppi20_in_8[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 8.
ppi20_in_9[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 9.
ppi21_in_0[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 0.
ppi21_in_1[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 1.
ppi21_in_10[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 10.
ppi21_in_11[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 11.
ppi21_in_12[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 12.
ppi21_in_13[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 13.
ppi21_in_14[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 14.
ppi21_in_15[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 15.
ppi21_in_16[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 16.
ppi21_in_17[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 17.
ppi21_in_18[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 18.
ppi21_in_19[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 19.
ppi21_in_2[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 2.
ppi21_in_20[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 20.
ppi21_in_21[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 21.
ppi21_in_22[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 22.
ppi21_in_23[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 23.
ppi21_in_24[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 24.
ppi21_in_25[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 25.
ppi21_in_26[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 26.
ppi21_in_27[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 27.
ppi21_in_28[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 28.
ppi21_in_29[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 29.
ppi21_in_3[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 3.
ppi21_in_30[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 30.
ppi21_in_31[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 31.
ppi21_in_4[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 4.
ppi21_in_5[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 5.

Name	Protocol	Type	Description
ppi21_in_6[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 6.
ppi21_in_7[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 7.
ppi21_in_8[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 8.
ppi21_in_9[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 9.
ppi22_in_0[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 0.
ppi22_in_1[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 1.
ppi22_in_10[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 10.
ppi22_in_11[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 11.
ppi22_in_12[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 12.
ppi22_in_13[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 13.
ppi22_in_14[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 14.
ppi22_in_15[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 15.
ppi22_in_16[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 16.
ppi22_in_17[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 17.
ppi22_in_18[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 18.
ppi22_in_19[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 19.
ppi22_in_2[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 2.
ppi22_in_20[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 20.
ppi22_in_21[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 21.
ppi22_in_22[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 22.
ppi22_in_23[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 23.
ppi22_in_24[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 24.
ppi22_in_25[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 25.
ppi22_in_26[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 26.
ppi22_in_27[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 27.

Name	Protocol	Type	Description
ppi22_in_28[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 28.
ppi22_in_29[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 29.
ppi22_in_3[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 3.
ppi22_in_30[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 30.
ppi22_in_31[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 31.
ppi22_in_4[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 4.
ppi22_in_5[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 5.
ppi22_in_6[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 6.
ppi22_in_7[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 7.
ppi22_in_8[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 8.
ppi22_in_9[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 9.
ppi23_in_0[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 0.
ppi23_in_1[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 1.
ppi23_in_10[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 10.
ppi23_in_11[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 11.
ppi23_in_12[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 12.
ppi23_in_13[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 13.
ppi23_in_14[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 14.
ppi23_in_15[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 15.
ppi23_in_16[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 16.
ppi23_in_17[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 17.
ppi23_in_18[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 18.
ppi23_in_19[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 19.

Name	Protocol	Type	Description
ppi23_in_2[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 2.
ppi23_in_20[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 20.
ppi23_in_21[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 21.
ppi23_in_22[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 22.
ppi23_in_23[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 23.
ppi23_in_24[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 24.
ppi23_in_25[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 25.
ppi23_in_26[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 26.
ppi23_in_27[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 27.
ppi23_in_28[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 28.
ppi23_in_29[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 29.
ppi23_in_3[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 3.
ppi23_in_30[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 30.
ppi23_in_31[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 31.
ppi23_in_4[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 4.
ppi23_in_5[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 5.
ppi23_in_6[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 6.
ppi23_in_7[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 7.
ppi23_in_8[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 8.
ppi23_in_9[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 9.
ppi24_in_0[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 0.
ppi24_in_1[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 1.
ppi24_in_10[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 10.

Name	Protocol	Type	Description
ppi24_in_11[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 11.
ppi24_in_12[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 12.
ppi24_in_13[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 13.
ppi24_in_14[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 14.
ppi24_in_15[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 15.
ppi24_in_16[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 16.
ppi24_in_17[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 17.
ppi24_in_18[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 18.
ppi24_in_19[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 19.
ppi24_in_2[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 2.
ppi24_in_20[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 20.
ppi24_in_21[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 21.
ppi24_in_22[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 22.
ppi24_in_23[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 23.
ppi24_in_24[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 24.
ppi24_in_25[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 25.
ppi24_in_26[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 26.
ppi24_in_27[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 27.
ppi24_in_28[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 28.
ppi24_in_29[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 29.
ppi24_in_3[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 3.
ppi24_in_30[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 30.
ppi24_in_31[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 31.

Name	Protocol	Type	Description
ppi24_in_4[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 4.
ppi24_in_5[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 5.
ppi24_in_6[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 6.
ppi24_in_7[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 7.
ppi24_in_8[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 8.
ppi24_in_9[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 9.
ppi25_in_0[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 0.
ppi25_in_1[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 1.
ppi25_in_10[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 10.
ppi25_in_11[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 11.
ppi25_in_12[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 12.
ppi25_in_13[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 13.
ppi25_in_14[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 14.
ppi25_in_15[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 15.
ppi25_in_16[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 16.
ppi25_in_17[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 17.
ppi25_in_18[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 18.
ppi25_in_19[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 19.
ppi25_in_2[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 2.
ppi25_in_20[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 20.
ppi25_in_21[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 21.
ppi25_in_22[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 22.
ppi25_in_23[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 23.

Name	Protocol	Type	Description
ppi25_in_24[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 24.
ppi25_in_25[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 25.
ppi25_in_26[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 26.
ppi25_in_27[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 27.
ppi25_in_28[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 28.
ppi25_in_29[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 29.
ppi25_in_3[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 3.
ppi25_in_30[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 30.
ppi25_in_31[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 31.
ppi25_in_4[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 4.
ppi25_in_5[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 5.
ppi25_in_6[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 6.
ppi25_in_7[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 7.
ppi25_in_8[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 8.
ppi25_in_9[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 9.
ppi26_in_0[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 0.
ppi26_in_1[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 1.
ppi26_in_10[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 10.
ppi26_in_11[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 11.
ppi26_in_12[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 12.
ppi26_in_13[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 13.
ppi26_in_14[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 14.
ppi26_in_15[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 15.
ppi26_in_16[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 16.
ppi26_in_17[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 17.
ppi26_in_18[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 18.
ppi26_in_19[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 19.
ppi26_in_2[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 2.
ppi26_in_20[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 20.

Name	Protocol	Type	Description
ppi26_in_21[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 21.
ppi26_in_22[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 22.
ppi26_in_23[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 23.
ppi26_in_24[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 24.
ppi26_in_25[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 25.
ppi26_in_26[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 26.
ppi26_in_27[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 27.
ppi26_in_28[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 28.
ppi26_in_29[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 29.
ppi26_in_3[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 3.
ppi26_in_30[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 30.
ppi26_in_31[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 31.
ppi26_in_4[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 4.
ppi26_in_5[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 5.
ppi26_in_6[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 6.
ppi26_in_7[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 7.
ppi26_in_8[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 8.
ppi26_in_9[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 9.
ppi27_in_0[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 0.
ppi27_in_1[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 1.
ppi27_in_10[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 10.
ppi27_in_11[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 11.
ppi27_in_12[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 12.
ppi27_in_13[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 13.
ppi27_in_14[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 14.
ppi27_in_15[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 15.
ppi27_in_16[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 16.
ppi27_in_17[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 17.
ppi27_in_18[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 18.
ppi27_in_19[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 19.
ppi27_in_2[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 2.
ppi27_in_20[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 20.
ppi27_in_21[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 21.
ppi27_in_22[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 22.
ppi27_in_23[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 23.
ppi27_in_24[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 24.
ppi27_in_25[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 25.
ppi27_in_26[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 26.
ppi27_in_27[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 27.

Name	Protocol	Type	Description
ppi27_in_28[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 28.
ppi27_in_29[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 29.
ppi27_in_3[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 3.
ppi27_in_30[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 30.
ppi27_in_31[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 31.
ppi27_in_4[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 4.
ppi27_in_5[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 5.
ppi27_in_6[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 6.
ppi27_in_7[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 7.
ppi27_in_8[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 8.
ppi27_in_9[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 9.
ppi28_in_0[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 0.
ppi28_in_1[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 1.
ppi28_in_10[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 10.
ppi28_in_11[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 11.
ppi28_in_12[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 12.
ppi28_in_13[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 13.
ppi28_in_14[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 14.
ppi28_in_15[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 15.
ppi28_in_16[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 16.
ppi28_in_17[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 17.
ppi28_in_18[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 18.
ppi28_in_19[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 19.
ppi28_in_2[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 2.
ppi28_in_20[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 20.
ppi28_in_21[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 21.
ppi28_in_22[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 22.
ppi28_in_23[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 23.
ppi28_in_24[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 24.
ppi28_in_25[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 25.
ppi28_in_26[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 26.
ppi28_in_27[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 27.
ppi28_in_28[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 28.
ppi28_in_29[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 29.
ppi28_in_3[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 3.
ppi28_in_30[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 30.
ppi28_in_31[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 31.
ppi28_in_4[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 4.
ppi28_in_5[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 5.

Name	Protocol	Type	Description
ppi28_in_6[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 6.
ppi28_in_7[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 7.
ppi28_in_8[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 8.
ppi28_in_9[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 9.
ppi29_in_0[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 0.
ppi29_in_1[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 1.
ppi29_in_10[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 10.
ppi29_in_11[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 11.
ppi29_in_12[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 12.
ppi29_in_13[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 13.
ppi29_in_14[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 14.
ppi29_in_15[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 15.
ppi29_in_16[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 16.
ppi29_in_17[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 17.
ppi29_in_18[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 18.
ppi29_in_19[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 19.
ppi29_in_2[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 2.
ppi29_in_20[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 20.
ppi29_in_21[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 21.
ppi29_in_22[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 22.
ppi29_in_23[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 23.
ppi29_in_24[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 24.
ppi29_in_25[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 25.
ppi29_in_26[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 26.
ppi29_in_27[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 27.
ppi29_in_28[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 28.
ppi29_in_29[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 29.
ppi29_in_3[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 3.
ppi29_in_30[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 30.
ppi29_in_31[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 31.
ppi29_in_4[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 4.
ppi29_in_5[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 5.
ppi29_in_6[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 6.
ppi29_in_7[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 7.
ppi29_in_8[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 8.
ppi29_in_9[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 9.
ppi30_in_0[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 0.
ppi30_in_1[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 1.

Name	Protocol	Type	Description
ppi30_in_10[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 10.
ppi30_in_11[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 11.
ppi30_in_12[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 12.
ppi30_in_13[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 13.
ppi30_in_14[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 14.
ppi30_in_15[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 15.
ppi30_in_16[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 16.
ppi30_in_17[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 17.
ppi30_in_18[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 18.
ppi30_in_19[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 19.
ppi30_in_2[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 2.
ppi30_in_20[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 20.
ppi30_in_21[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 21.
ppi30_in_22[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 22.
ppi30_in_23[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 23.
ppi30_in_24[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 24.
ppi30_in_25[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 25.
ppi30_in_26[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 26.
ppi30_in_27[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 27.
ppi30_in_28[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 28.
ppi30_in_29[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 29.
ppi30_in_3[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 3.
ppi30_in_30[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 30.

Name	Protocol	Type	Description
ppi30_in_31[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 31.
ppi30_in_4[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 4.
ppi30_in_5[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 5.
ppi30_in_6[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 6.
ppi30_in_7[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 7.
ppi30_in_8[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 8.
ppi30_in_9[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 9.
ppi31_in_0[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 0.
ppi31_in_1[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 1.
ppi31_in_10[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 10.
ppi31_in_11[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 11.
ppi31_in_12[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 12.
ppi31_in_13[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 13.
ppi31_in_14[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 14.
ppi31_in_15[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 15.
ppi31_in_16[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 16.
ppi31_in_17[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 17.
ppi31_in_18[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 18.
ppi31_in_19[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 19.
ppi31_in_2[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 2.
ppi31_in_20[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 20.
ppi31_in_21[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 21.
ppi31_in_22[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 22.
ppi31_in_23[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 23.
ppi31_in_24[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 24.
ppi31_in_25[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 25.
ppi31_in_26[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 26.
ppi31_in_27[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 27.
ppi31_in_28[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 28.
ppi31_in_29[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 29.
ppi31_in_3[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 3.
ppi31_in_30[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 30.
ppi31_in_31[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 31.
ppi31_in_4[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 4.

Name	Protocol	Type	Description
ppi31_in_5[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 5.
ppi31_in_6[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 6.
ppi31_in_7[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 7.
ppi31_in_8[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 8.
ppi31_in_9[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 9.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_0[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 0
redistributor_1[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 1.
redistributor_10[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 10.
redistributor_11[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 11.
redistributor_12[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 12.
redistributor_13[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 13.
redistributor_14[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 14.
redistributor_15[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 15.
redistributor_16[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 16.
redistributor_17[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 17.
redistributor_18[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 18.
redistributor_19[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 19.
redistributor_2[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 2.
redistributor_20[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 20
redistributor_21[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 21.
redistributor_22[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 22.
redistributor_23[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 23
redistributor_24[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 24.
redistributor_25[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 25.
redistributor_26[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 26.
redistributor_27[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 27.
redistributor_28[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 28.
redistributor_29[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 29.
redistributor_3[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 3.
redistributor_30[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 30.
redistributor_31[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 31.
redistributor_4[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 4.
redistributor_5[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 5.
redistributor_6[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 6.
redistributor_7[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 7.
redistributor_8[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 8.
redistributor_9[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 9.

Name	Protocol	Type	Description
reset	Signal	Slave	Reset.
spi_in[988]	Signal	Slave	Shared peripheral interrupts.
wake_request_0[8]	Signal	Master	Power management outputs of cluster 0.
wake_request_1[8]	Signal	Master	Power management outputs of cluster 1.
wake_request_10[8]	Signal	Master	Power management outputs of cluster 10.
wake_request_11[8]	Signal	Master	Power management outputs of cluster 11.
wake_request_12[8]	Signal	Master	Power management outputs of cluster 12.
wake_request_13[8]	Signal	Master	Power management outputs of cluster 13.
wake_request_14[8]	Signal	Master	Power management outputs of cluster 14.
wake_request_15[8]	Signal	Master	Power management outputs of cluster 15.
wake_request_16[8]	Signal	Master	Power management outputs of cluster 16.
wake_request_17[8]	Signal	Master	Power management outputs of cluster 17.
wake_request_18[8]	Signal	Master	Power management outputs of cluster 18.
wake_request_19[8]	Signal	Master	Power management outputs of cluster 19.
wake_request_2[8]	Signal	Master	Power management outputs of cluster 2.
wake_request_20[8]	Signal	Master	Power management outputs of cluster 20.
wake_request_21[8]	Signal	Master	Power management outputs of cluster 21.
wake_request_22[8]	Signal	Master	Power management outputs of cluster 22.
wake_request_23[8]	Signal	Master	Power management outputs of cluster 23.
wake_request_24[8]	Signal	Master	Power management outputs of cluster 24.
wake_request_25[8]	Signal	Master	Power management outputs of cluster 25.
wake_request_26[8]	Signal	Master	Power management outputs of cluster 26.
wake_request_27[8]	Signal	Master	Power management outputs of cluster 27.
wake_request_28[8]	Signal	Master	Power management outputs of cluster 28.
wake_request_29[8]	Signal	Master	Power management outputs of cluster 29.
wake_request_3[8]	Signal	Master	Power management outputs of cluster 3.
wake_request_30[8]	Signal	Master	Power management outputs of cluster 30.
wake_request_31[8]	Signal	Master	Power management outputs of cluster 31.
wake_request_4[8]	Signal	Master	Power management outputs of cluster 4.
wake_request_5[8]	Signal	Master	Power management outputs of cluster 5.
wake_request_6[8]	Signal	Master	Power management outputs of cluster 6.
wake_request_7[8]	Signal	Master	Power management outputs of cluster 7.
wake_request_8[8]	Signal	Master	Power management outputs of cluster 8.
wake_request_9[8]	Signal	Master	Power management outputs of cluster 9.

Parameters for GIC500

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: bool. Default value: 0x0.

common-vPE-table-affinity

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.

Type: `string`. Default value: `""`.

cpus_per_cluster_0

Number of cores within cluster 0.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_1

Number of cores within cluster 1.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_10

Number of cores within cluster 10.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_11

Number of cores within cluster 11.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_12

Number of cores within cluster 12.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_13

Number of cores within cluster 13.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_14

Number of cores within cluster 14.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_15

Number of cores within cluster 15.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_16

Number of cores within cluster 16.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_17

Number of cores within cluster 17.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_18

Number of cores within cluster 18.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_19

Number of cores within cluster 19.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_2`

Number of cores within cluster 2.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_20`

Number of cores within cluster 20.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_21`

Number of cores within cluster 21.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_22`

Number of cores within cluster 22.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_23`

Number of cores within cluster 23.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_24`

Number of cores within cluster 24.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_25`

Number of cores within cluster 25.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_26`

Number of cores within cluster 26.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_27`

Number of cores within cluster 27.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_28`

Number of cores within cluster 28.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_29`

Number of cores within cluster 29.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_3`

Number of cores within cluster 3.

Type: `int`. Default value: `0x1`.

`cpus_per_cluster_30`

Number of cores within cluster 30.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_31

Number of cores within cluster 31.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_4

Number of cores within cluster 4.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_5

Number of cores within cluster 5.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_6

Number of cores within cluster 6.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_7

Number of cores within cluster 7.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_8

Number of cores within cluster 8.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_9

Number of cores within cluster 9.

Type: `int`. Default value: `0x1`.

delay-ITS-accesses

Delay accesses from the ITS until GICR_SYNCRR is read.

Type: `bool`. Default value: `0x1`.

delay-redistributor-accesses

Delay memory accesses from the redistributor until GICR_SYNCRR is read.

Type: `bool`. Default value: `0x1`.

enable_protocol_checking

Enable/disable protocol checking at cpu interface.

Type: `bool`. Default value: `0x0`.

enabled

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

GICD_ITARGETSR-RAZWI

If true, the GICD_ITARGETS registers are RAZ/WI.

Type: `bool`. Default value: `0x0`.

has-two-security-states

If true, has two security states.

Type: `bool`. Default value: `0x1`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `int`. Default value: `0x0`.

ITS-threaded-command-queue

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

num_clusters

Number of implemented affinity level1 clusters.

Type: `int`. Default value: `0x1`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

redistributor-threaded-sync

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

reg-base

GIC500 base address.

Type: `int`. Default value: `0x2c010000`.

SPI-count

Number of SPIs that are implemented.

Type: `int`. Default value: `0xe0`.

using-generated-memorymap

Use the generated memorymap for the GIC500 and warn if superfluous parameters are passed.

Type: `bool`. Default value: `0x1`.

wakeup-on-reset

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.

Type: `bool`. Default value: `0x0`.

4.10.28 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 4-395: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC500_Filter contains the following CADI targets:

- GIC500

GIC500_Filter contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3InterruptTranslationService](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)

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 Arm®v8-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 the [MSIRewriter](#) component.

MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

Table 4-396: pvb_{bus}_s port

PVBus attribute	Bits used	Property encoded	Notes
MasterID	Bits[31:0]	Master ID that invoked the register access	-
ExtendedID	-	-	Not used.
UserFlags	-	-	Not used.



The pvb_{bus}_m port does not use MasterID, ExtendedID, Or UserFlags.

Ports for GIC500_Filter

Table 4-397: Ports

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	Disable some SPIs signal.
cpu_active_0[8]	Signal	Slave	cpu_active pins of cluster 0.
cpu_active_1[8]	Signal	Slave	cpu_active pins of cluster 1.
cpu_active_10[8]	Signal	Slave	cpu_active pins of cluster 10.
cpu_active_11[8]	Signal	Slave	cpu_active pins of cluster 11.
cpu_active_12[8]	Signal	Slave	cpu_active pins of cluster 12.
cpu_active_13[8]	Signal	Slave	cpu_active pins of cluster 13.
cpu_active_14[8]	Signal	Slave	cpu_active pins of cluster 14.
cpu_active_15[8]	Signal	Slave	cpu_active pins of cluster 15.
cpu_active_16[8]	Signal	Slave	cpu_active pins of cluster 16.
cpu_active_17[8]	Signal	Slave	cpu_active pins of cluster 17.
cpu_active_18[8]	Signal	Slave	cpu_active pins of cluster 18.
cpu_active_19[8]	Signal	Slave	cpu_active pins of cluster 19.
cpu_active_2[8]	Signal	Slave	cpu_active pins of cluster 2.
cpu_active_20[8]	Signal	Slave	cpu_active pins of cluster 20.
cpu_active_21[8]	Signal	Slave	cpu_active pins of cluster 21.
cpu_active_22[8]	Signal	Slave	cpu_active pins of cluster 22.
cpu_active_23[8]	Signal	Slave	cpu_active pins of cluster 23.
cpu_active_24[8]	Signal	Slave	cpu_active pins of cluster 24.
cpu_active_25[8]	Signal	Slave	cpu_active pins of cluster 25.
cpu_active_26[8]	Signal	Slave	cpu_active pins of cluster 26.
cpu_active_27[8]	Signal	Slave	cpu_active pins of cluster 27.
cpu_active_28[8]	Signal	Slave	cpu_active pins of cluster 28.
cpu_active_29[8]	Signal	Slave	cpu_active pins of cluster 29.
cpu_active_3[8]	Signal	Slave	cpu_active pins of cluster 3.
cpu_active_30[8]	Signal	Slave	cpu_active pins of cluster 30.

Name	Protocol	Type	Description
cpu_active_31[8]	Signal	Slave	cpu_active pins of cluster 31.
cpu_active_4[8]	Signal	Slave	cpu_active pins of cluster 4.
cpu_active_5[8]	Signal	Slave	cpu_active pins of cluster 5.
cpu_active_6[8]	Signal	Slave	cpu_active pins of cluster 6.
cpu_active_7[8]	Signal	Slave	cpu_active pins of cluster 7.
cpu_active_8[8]	Signal	Slave	cpu_active pins of cluster 8.
cpu_active_9[8]	Signal	Slave	cpu_active pins of cluster 9.
po_reset	Signal	Slave	Power on reset.
ppi16_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 0.
ppi16_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 1.
ppi16_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 10.
ppi16_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 11.
ppi16_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 12.
ppi16_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 13.
ppi16_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 14.
ppi16_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 15.
ppi16_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 16.
ppi16_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 17.
ppi16_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 18.
ppi16_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 19.
ppi16_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 2.
ppi16_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 20.
ppi16_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 21.
ppi16_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 22.
ppi16_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 23.
ppi16_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 24.
ppi16_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 25.
ppi16_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 26.
ppi16_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 27.
ppi16_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 28.
ppi16_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 29.
ppi16_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 3.
ppi16_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 30.
ppi16_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 31.
ppi16_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 4.
ppi16_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 5.
ppi16_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 6.
ppi16_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 7.
ppi16_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 8.

Name	Protocol	Type	Description
ppi16_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 9.
ppi17_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 0.
ppi17_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 1.
ppi17_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 10.
ppi17_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 11.
ppi17_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 12.
ppi17_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 13.
ppi17_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 14.
ppi17_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 15.
ppi17_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 16.
ppi17_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 17.
ppi17_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 18.
ppi17_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 19.
ppi17_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 2.
ppi17_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 20.
ppi17_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 21.
ppi17_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 22.
ppi17_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 23.
ppi17_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 24.
ppi17_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 25.
ppi17_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 26.
ppi17_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 27.
ppi17_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 28.
ppi17_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 29.
ppi17_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 3.
ppi17_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 30.
ppi17_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 31.
ppi17_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 4.
ppi17_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 5.
ppi17_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 6.
ppi17_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 7.
ppi17_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 8.
ppi17_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 9.
ppi18_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 0.
ppi18_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 1.
ppi18_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 10.
ppi18_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 11.
ppi18_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 12.
ppi18_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 13.

Name	Protocol	Type	Description
ppi18_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 14.
ppi18_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 15.
ppi18_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 16.
ppi18_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 17.
ppi18_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 18.
ppi18_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 19.
ppi18_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 2.
ppi18_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 20.
ppi18_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 21.
ppi18_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 22.
ppi18_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 23.
ppi18_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 24.
ppi18_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 25.
ppi18_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 26.
ppi18_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 27.
ppi18_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 28.
ppi18_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 29.
ppi18_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 3.
ppi18_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 30.
ppi18_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 31.
ppi18_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 4.
ppi18_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 5.
ppi18_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 6.
ppi18_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 7.
ppi18_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 8.
ppi18_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 9.
ppi19_in_0[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 0.
ppi19_in_1[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 1.
ppi19_in_10[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 10.
ppi19_in_11[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 11.
ppi19_in_12[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 12.
ppi19_in_13[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 13.
ppi19_in_14[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 14.
ppi19_in_15[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 15.
ppi19_in_16[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 16.
ppi19_in_17[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 17.
ppi19_in_18[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 18.
ppi19_in_19[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 19.
ppi19_in_2[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 2.

Name	Protocol	Type	Description
ppi19_in_20[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 20.
ppi19_in_21[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 21.
ppi19_in_22[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 22.
ppi19_in_23[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 23.
ppi19_in_24[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 24.
ppi19_in_25[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 25.
ppi19_in_26[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 26.
ppi19_in_27[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 27.
ppi19_in_28[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 28.
ppi19_in_29[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 29.
ppi19_in_3[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 3.
ppi19_in_30[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 30.
ppi19_in_31[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 31.
ppi19_in_4[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 4.
ppi19_in_5[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 5.
ppi19_in_6[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 6.
ppi19_in_7[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 7.
ppi19_in_8[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 8.
ppi19_in_9[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 9.
ppi20_in_0[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 0.
ppi20_in_1[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 1.
ppi20_in_10[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 10.
ppi20_in_11[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 11.
ppi20_in_12[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 12.
ppi20_in_13[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 13.
ppi20_in_14[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 14.
ppi20_in_15[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 15.
ppi20_in_16[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 16.
ppi20_in_17[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 17.
ppi20_in_18[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 18.
ppi20_in_19[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 19.
ppi20_in_2[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 2.
ppi20_in_20[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 20.
ppi20_in_21[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 21.
ppi20_in_22[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 22.
ppi20_in_23[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 23.
ppi20_in_24[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 24.
ppi20_in_25[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 25.
ppi20_in_26[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 26.

Name	Protocol	Type	Description
ppi20_in_27[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 27.
ppi20_in_28[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 28.
ppi20_in_29[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 29.
ppi20_in_3[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 3.
ppi20_in_30[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 30.
ppi20_in_31[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 31.
ppi20_in_4[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 4.
ppi20_in_5[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 5.
ppi20_in_6[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 6.
ppi20_in_7[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 7.
ppi20_in_8[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 8.
ppi20_in_9[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 9.
ppi21_in_0[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 0.
ppi21_in_1[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 1.
ppi21_in_10[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 10.
ppi21_in_11[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 11.
ppi21_in_12[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 12.
ppi21_in_13[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 13.
ppi21_in_14[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 14.
ppi21_in_15[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 15.
ppi21_in_16[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 16.
ppi21_in_17[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 17.
ppi21_in_18[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 18.
ppi21_in_19[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 19.
ppi21_in_2[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 2.
ppi21_in_20[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 20.
ppi21_in_21[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 21.
ppi21_in_22[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 22.
ppi21_in_23[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 23.
ppi21_in_24[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 24.
ppi21_in_25[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 25.
ppi21_in_26[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 26.
ppi21_in_27[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 27.
ppi21_in_28[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 28.
ppi21_in_29[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 29.
ppi21_in_3[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 3.
ppi21_in_30[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 30.
ppi21_in_31[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 31.
ppi21_in_4[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 4.

Name	Protocol	Type	Description
ppi21_in_5[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 5.
ppi21_in_6[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 6.
ppi21_in_7[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 7.
ppi21_in_8[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 8.
ppi21_in_9[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 9.
ppi22_in_0[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 0.
ppi22_in_1[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 1.
ppi22_in_10[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 10.
ppi22_in_11[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 11.
ppi22_in_12[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 12.
ppi22_in_13[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 13.
ppi22_in_14[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 14.
ppi22_in_15[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 15.
ppi22_in_16[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 16.
ppi22_in_17[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 17.
ppi22_in_18[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 18.
ppi22_in_19[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 19.
ppi22_in_2[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 2.
ppi22_in_20[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 20.
ppi22_in_21[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 21.
ppi22_in_22[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 22.
ppi22_in_23[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 23.
ppi22_in_24[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 24.
ppi22_in_25[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 25.
ppi22_in_26[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 26.

Name	Protocol	Type	Description
ppi22_in_27[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 27.
ppi22_in_28[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 28.
ppi22_in_29[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 29.
ppi22_in_3[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 3.
ppi22_in_30[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 30.
ppi22_in_31[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 31.
ppi22_in_4[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 4.
ppi22_in_5[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 5.
ppi22_in_6[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 6.
ppi22_in_7[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 7.
ppi22_in_8[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 8.
ppi22_in_9[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 9.
ppi23_in_0[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 0.
ppi23_in_1[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 1.
ppi23_in_10[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 10.
ppi23_in_11[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 11.
ppi23_in_12[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 12.
ppi23_in_13[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 13.
ppi23_in_14[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 14.
ppi23_in_15[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 15.
ppi23_in_16[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 16.
ppi23_in_17[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 17.
ppi23_in_18[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 18.

Name	Protocol	Type	Description
ppi23_in_19[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 19.
ppi23_in_2[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 2.
ppi23_in_20[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 20.
ppi23_in_21[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 21.
ppi23_in_22[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 22.
ppi23_in_23[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 23.
ppi23_in_24[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 24.
ppi23_in_25[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 25.
ppi23_in_26[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 26.
ppi23_in_27[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 27.
ppi23_in_28[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 28.
ppi23_in_29[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 29.
ppi23_in_3[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 3.
ppi23_in_30[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 30.
ppi23_in_31[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 31.
ppi23_in_4[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 4.
ppi23_in_5[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 5.
ppi23_in_6[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 6.
ppi23_in_7[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 7.
ppi23_in_8[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 8.
ppi23_in_9[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 9.
ppi24_in_0[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 0.
ppi24_in_1[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 1.

Name	Protocol	Type	Description
ppi24_in_10[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 10.
ppi24_in_11[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 11.
ppi24_in_12[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 12.
ppi24_in_13[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 13.
ppi24_in_14[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 14.
ppi24_in_15[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 15.
ppi24_in_16[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 16.
ppi24_in_17[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 17.
ppi24_in_18[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 18.
ppi24_in_19[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 19.
ppi24_in_2[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 2.
ppi24_in_20[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 20.
ppi24_in_21[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 21.
ppi24_in_22[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 22.
ppi24_in_23[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 23.
ppi24_in_24[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 24.
ppi24_in_25[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 25.
ppi24_in_26[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 26.
ppi24_in_27[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 27.
ppi24_in_28[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 28.
ppi24_in_29[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 29.
ppi24_in_3[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 3.
ppi24_in_30[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 30.

Name	Protocol	Type	Description
ppi24_in_31[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 31.
ppi24_in_4[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 4.
ppi24_in_5[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 5.
ppi24_in_6[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 6.
ppi24_in_7[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 7.
ppi24_in_8[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 8.
ppi24_in_9[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 9.
ppi25_in_0[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 0.
ppi25_in_1[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 1.
ppi25_in_10[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 10.
ppi25_in_11[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 11.
ppi25_in_12[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 12.
ppi25_in_13[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 13.
ppi25_in_14[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 14.
ppi25_in_15[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 15.
ppi25_in_16[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 16.
ppi25_in_17[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 17.
ppi25_in_18[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 18.
ppi25_in_19[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 19.
ppi25_in_2[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 2.
ppi25_in_20[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 20.
ppi25_in_21[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 21.
ppi25_in_22[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 22.

Name	Protocol	Type	Description
ppi25_in_23[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 23.
ppi25_in_24[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 24.
ppi25_in_25[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 25.
ppi25_in_26[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 26.
ppi25_in_27[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 27.
ppi25_in_28[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 28.
ppi25_in_29[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 29.
ppi25_in_3[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 3.
ppi25_in_30[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 30.
ppi25_in_31[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 31.
ppi25_in_4[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 4.
ppi25_in_5[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 5.
ppi25_in_6[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 6.
ppi25_in_7[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 7.
ppi25_in_8[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 8.
ppi25_in_9[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 9.
ppi26_in_0[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 0.
ppi26_in_1[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 1.
ppi26_in_10[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 10.
ppi26_in_11[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 11.
ppi26_in_12[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 12.
ppi26_in_13[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 13.
ppi26_in_14[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 14.
ppi26_in_15[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 15.
ppi26_in_16[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 16.
ppi26_in_17[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 17.
ppi26_in_18[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 18.
ppi26_in_19[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 19.

Name	Protocol	Type	Description
ppi26_in_2[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 2.
ppi26_in_20[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 20.
ppi26_in_21[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 21.
ppi26_in_22[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 22.
ppi26_in_23[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 23.
ppi26_in_24[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 24.
ppi26_in_25[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 25.
ppi26_in_26[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 26.
ppi26_in_27[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 27.
ppi26_in_28[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 28.
ppi26_in_29[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 29.
ppi26_in_3[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 3.
ppi26_in_30[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 30.
ppi26_in_31[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 31.
ppi26_in_4[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 4.
ppi26_in_5[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 5.
ppi26_in_6[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 6.
ppi26_in_7[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 7.
ppi26_in_8[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 8.
ppi26_in_9[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 9.
ppi27_in_0[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 0.
ppi27_in_1[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 1.
ppi27_in_10[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 10.
ppi27_in_11[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 11.
ppi27_in_12[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 12.
ppi27_in_13[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 13.
ppi27_in_14[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 14.
ppi27_in_15[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 15.
ppi27_in_16[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 16.
ppi27_in_17[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 17.
ppi27_in_18[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 18.
ppi27_in_19[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 19.
ppi27_in_2[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 2.
ppi27_in_20[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 20.
ppi27_in_21[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 21.
ppi27_in_22[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 22.
ppi27_in_23[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 23.
ppi27_in_24[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 24.
ppi27_in_25[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 25.

Name	Protocol	Type	Description
ppi27_in_26[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 26.
ppi27_in_27[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 27.
ppi27_in_28[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 28.
ppi27_in_29[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 29.
ppi27_in_3[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 3.
ppi27_in_30[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 30.
ppi27_in_31[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 31.
ppi27_in_4[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 4.
ppi27_in_5[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 5.
ppi27_in_6[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 6.
ppi27_in_7[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 7.
ppi27_in_8[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 8.
ppi27_in_9[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 9.
ppi28_in_0[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 0.
ppi28_in_1[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 1.
ppi28_in_10[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 10.
ppi28_in_11[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 11.
ppi28_in_12[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 12.
ppi28_in_13[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 13.
ppi28_in_14[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 14.
ppi28_in_15[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 15.
ppi28_in_16[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 16.
ppi28_in_17[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 17.
ppi28_in_18[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 18.
ppi28_in_19[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 19.
ppi28_in_2[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 2.
ppi28_in_20[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 20.
ppi28_in_21[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 21.
ppi28_in_22[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 22.
ppi28_in_23[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 23.
ppi28_in_24[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 24.
ppi28_in_25[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 25.
ppi28_in_26[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 26.
ppi28_in_27[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 27.
ppi28_in_28[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 28.
ppi28_in_29[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 29.
ppi28_in_3[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 3.
ppi28_in_30[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 30.
ppi28_in_31[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 31.

Name	Protocol	Type	Description
ppi28_in_4[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 4.
ppi28_in_5[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 5.
ppi28_in_6[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 6.
ppi28_in_7[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 7.
ppi28_in_8[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 8.
ppi28_in_9[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 9.
ppi29_in_0[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 0.
ppi29_in_1[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 1.
ppi29_in_10[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 10.
ppi29_in_11[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 11.
ppi29_in_12[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 12.
ppi29_in_13[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 13.
ppi29_in_14[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 14.
ppi29_in_15[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 15.
ppi29_in_16[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 16.
ppi29_in_17[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 17.
ppi29_in_18[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 18.
ppi29_in_19[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 19.
ppi29_in_2[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 2.
ppi29_in_20[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 20.
ppi29_in_21[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 21.
ppi29_in_22[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 22.
ppi29_in_23[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 23.
ppi29_in_24[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 24.
ppi29_in_25[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 25.
ppi29_in_26[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 26.
ppi29_in_27[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 27.
ppi29_in_28[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 28.
ppi29_in_29[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 29.
ppi29_in_3[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 3.
ppi29_in_30[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 30.
ppi29_in_31[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 31.
ppi29_in_4[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 4.
ppi29_in_5[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 5.
ppi29_in_6[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 6.
ppi29_in_7[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 7.
ppi29_in_8[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 8.
ppi29_in_9[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 9.

Name	Protocol	Type	Description
ppi30_in_0[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 0.
ppi30_in_1[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 1.
ppi30_in_10[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 10.
ppi30_in_11[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 11.
ppi30_in_12[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 12.
ppi30_in_13[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 13.
ppi30_in_14[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 14.
ppi30_in_15[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 15.
ppi30_in_16[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 16.
ppi30_in_17[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 17.
ppi30_in_18[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 18.
ppi30_in_19[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 19.
ppi30_in_2[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 2.
ppi30_in_20[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 20.
ppi30_in_21[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 21.
ppi30_in_22[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 22.
ppi30_in_23[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 23.
ppi30_in_24[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 24.
ppi30_in_25[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 25.
ppi30_in_26[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 26.
ppi30_in_27[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 27.
ppi30_in_28[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 28.
ppi30_in_29[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 29.

Name	Protocol	Type	Description
ppi30_in_3[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 3.
ppi30_in_30[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 30.
ppi30_in_31[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 31.
ppi30_in_4[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 4.
ppi30_in_5[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 5.
ppi30_in_6[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 6.
ppi30_in_7[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 7.
ppi30_in_8[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 8.
ppi30_in_9[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 9.
ppi31_in_0[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 0.
ppi31_in_1[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 1.
ppi31_in_10[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 10.
ppi31_in_11[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 11.
ppi31_in_12[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 12.
ppi31_in_13[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 13.
ppi31_in_14[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 14.
ppi31_in_15[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 15.
ppi31_in_16[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 16.
ppi31_in_17[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 17.
ppi31_in_18[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 18.
ppi31_in_19[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 19.
ppi31_in_2[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 2.
ppi31_in_20[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 20.
ppi31_in_21[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 21.
ppi31_in_22[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 22.
ppi31_in_23[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 23.
ppi31_in_24[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 24.
ppi31_in_25[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 25.
ppi31_in_26[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 26.
ppi31_in_27[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 27.
ppi31_in_28[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 28.
ppi31_in_29[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 29.
ppi31_in_3[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 3.

Name	Protocol	Type	Description
ppi31_in_30[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 30.
ppi31_in_31[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 31.
ppi31_in_4[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 4.
ppi31_in_5[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 5.
ppi31_in_6[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 6.
ppi31_in_7[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 7.
ppi31_in_8[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 8.
ppi31_in_9[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 9.
pvbus_filtermiss_m	PVBus	Master	passthrough for transactions not targetting one of the pages associated with the IRI.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Reset.
spi_in[988]	Signal	Slave	Shared peripheral interrupts.
wake_request_0[8]	Signal	Master	Power management outputs of cluster 0.
wake_request_1[8]	Signal	Master	Power management outputs of cluster 1.
wake_request_10[8]	Signal	Master	Power management outputs of cluster 10.
wake_request_11[8]	Signal	Master	Power management outputs of cluster 11.
wake_request_12[8]	Signal	Master	Power management outputs of cluster 12.
wake_request_13[8]	Signal	Master	Power management outputs of cluster 13.
wake_request_14[8]	Signal	Master	Power management outputs of cluster 14.
wake_request_15[8]	Signal	Master	Power management outputs of cluster 15.
wake_request_16[8]	Signal	Master	Power management outputs of cluster 16.
wake_request_17[8]	Signal	Master	Power management outputs of cluster 17.
wake_request_18[8]	Signal	Master	Power management outputs of cluster 18.
wake_request_19[8]	Signal	Master	Power management outputs of cluster 19.
wake_request_2[8]	Signal	Master	Power management outputs of cluster 2.
wake_request_20[8]	Signal	Master	Power management outputs of cluster 20.
wake_request_21[8]	Signal	Master	Power management outputs of cluster 21.
wake_request_22[8]	Signal	Master	Power management outputs of cluster 22.
wake_request_23[8]	Signal	Master	Power management outputs of cluster 23.
wake_request_24[8]	Signal	Master	Power management outputs of cluster 24.
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.

Name	Protocol	Type	Description
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

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`. Default value: `0x0`.

common-vPE-table-affinity

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.

Type: `string`. Default value: `""`.

cpus_per_cluster_0

Number of cores within cluster 0.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_1

Number of cores within cluster 1.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_10

Number of cores within cluster 10.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_11

Number of cores within cluster 11.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_12

Number of cores within cluster 12.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_13

Number of cores within cluster 13.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_14

Number of cores within cluster 14.

Type: `int`. Default value: `0x1`.

cpus_per_cluster_15

Number of cores within cluster 15.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_16

Number of cores within cluster 16.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_17

Number of cores within cluster 17.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_18

Number of cores within cluster 18.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_19

Number of cores within cluster 19.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_2

Number of cores within cluster 2.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_20

Number of cores within cluster 20.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_21

Number of cores within cluster 21.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_22

Number of cores within cluster 22.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_23

Number of cores within cluster 23.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_24

Number of cores within cluster 24.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_25

Number of cores within cluster 25.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_26

Number of cores within cluster 26.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_27

Number of cores within cluster 27.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_28

Number of cores within cluster 28.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_29

Number of cores within cluster 29.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_3

Number of cores within cluster 3.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_30

Number of cores within cluster 30.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_31

Number of cores within cluster 31.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_4

Number of cores within cluster 4.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_5

Number of cores within cluster 5.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_6

Number of cores within cluster 6.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_7

Number of cores within cluster 7.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_8

Number of cores within cluster 8.
Type: `int`. Default value: `0x1`.

cpus_per_cluster_9

Number of cores within cluster 9.
Type: `int`. Default value: `0x1`.

delay-ITS-accesses

Delay accesses from the ITS until GICR_SYNC is read.
Type: `bool`. Default value: `0x1`.

delay-redistributor-accesses

Delay memory accesses from the redistributor until GICR_SYNCRR is read.

Type: `bool`. Default value: `0x1`.

enable_protocol_checking

Enable/disable protocol checking at cpu interface.

Type: `bool`. Default value: `0x0`.

enabled

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

GICD_ITARGETSR-RAZWI

If true, the GICD_ITARGETSR registers are RAZ/WI.

Type: `bool`. Default value: `0x0`.

has-two-security-states

If true, has two security states.

Type: `bool`. Default value: `0x1`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

Type: `int`. Default value: `0x0`.

ITS-threaded-command-queue

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

num_clusters

Number of implemented affinity level1 clusters.

Type: `int`. Default value: `0x1`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

redistributor-threaded-sync

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

reg-base

GIC500 base address.

Type: `int`. Default value: `0x2c010000`.

SPI-count

Number of SPIs that are implemented.
Type: `int`. Default value: `0xe0`.

using-generated-memorymap

Use the generated memorymap for the GIC500 and warn if superfluous parameters are passed.
Type: `bool`. Default value: `0x1`.

wakeup-on-reset

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.
Type: `bool`. Default value: `0x0`.

4.10.29 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 4-398: IP revisions support

Revision	Quality level
r1p6	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC600 contains the following CADI targets:

- GIC600

GIC600 contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3InterruptTranslationService](#)
- [GICv3ProtocolChecker](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)

About GIC600

GIC600 and GIC600_Filter are minimal models of an Arm® GIC-600 Generic Interrupt Controller, suitable for single-chip or multichip systems, where multichip operation has been tested, with the following exceptions:

- LPI/LPI command is implemented but untested.
- Power operation is unimplemented.

They provide a simple configuration interface that allows designers to introduce GIC600-like functionality to their systems, while only implementing the architectural behavior, as defined by the GICv3 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC600 and GIC600_Filter have a `pvbus_s` port for register accesses and a `pvbus_m` port for the LPI-related traffic from redistributors and the ITS. The `pvbus_m` port is also used to compose multichip operation.

In addition, the GIC600_Filter variant has a `pvbus_filtermiss_m` port, to which any transactions on the `pvbus_s` port and not directed to a 4K page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC600 variant.

It is recommended to use the GIC600 variant in most cases.

Also, the RAS register is implemented and various RAS errors are reported through status registers. Not all the RAS errors are available because Fast Models does not model errors that can cause ECC errors. Fast Models supports error record 0 and error record 13+.



Note

- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to `stderr` the memory map of any GICv3 or later models that are included in the platform being run.
- For writes to `GITS_TRANSLATE64R`, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see the [MSIRewriter](#) component.

MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

Table 4-399: `pvbus_s` port

PVBus attribute	Bits used	Property encoded	Notes
MasterID	Bits[31:0]	Master ID that invoked the register access	-
ExtendedID	-	-	Not used.
UserFlags	-	-	Not used.



The `pvbush_m` port does not use `MasterID`, `ExtendedID`, or `UserFlags`.

Ports for GIC600

Table 4-400: 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	Resets.
<code>ppi_in_<n>[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for <code>cpu <n></code> , where <code>n</code> is in the range 0-255.
<code>pvbush_m</code>	PVBus	Master	Memory bus out: transactions generated by the IRI.
<code>pvbush_s</code>	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
<code>redistributor_m[256]</code>	GICv3Comms	Master	Input from and output to CPU interface.
<code>reset</code>	Signal	Slave	Resets.
<code>spi_in[960]</code>	Signal	Slave	Shared peripheral interrupts.
<code>wake_request[256]</code>	Signal	Master	Power management outputs.

Parameters for GIC600

affinity-width

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`. Default value: "4.8.8.8".

chip-id

Chip ID when multichip operation is enabled.

Type: `int`. Default value: `0x0`.

chip-select-affinity-level

Affinity level 2 or 3 can be used for chip select.

Type: `int`. Default value: `0x3`.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`. Default value: `0x0`.

common-vPE-table-affinity

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.

Type: `string`. Default value: "".

CPU-affinities

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.

Type: `string`. Default value: `""`.

CPU-affinities-file

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.

Type: `string`. Default value: `""`.

direct-lpi-support

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: `0x0`.

DS-behaviour

GICD_CTLR.DS field behaviour: 0:RAZ/WI, 1:RAO/WI, 2:RW.

Type: `int`. Default value: `0x2`.

enabled

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

gict-allow-ns-reset

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.

Type: `bool`. Default value: `0x1`.

IIDR

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.

Type: `int`. Default value: `0x0`.

ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x8`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-ID-bits

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

Type: `int`. Default value: `0x0`.

max-cores-supported-by-GCI

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`. Default value: `0x8`.

max-pe-on-chip

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if `enable-multichip-operation` is set.

Type: `bool`. Default value: `0x1`.

PPI-count

Selects the number of PPI available for each PE as 8(id22-27,29,30), 12(id 20-31) or 16(id 16-31).

Type: `int`. Default value: `0x10`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

RAS-CFI-support

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-FI-support

If true, fault handling interrupt is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-UE-support

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-UI-support

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

redistributor-group

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.

Type: `string`. Default value: `""`.

redistributor-group-file

File path to redistributor grouping information with affinity as JSON. The file uses the same format as "redistributor-group" parameter.

Type: `string`. Default value: `""`.

redistributor-power-managed-by-pwrr

GIC600 dedistributor power management is done by updating GICR_PWRR register.
Type: `bool`. Default value: `0x1`.

reg-base

GIC-600 base address.
Type: `int`. Default value: `0x2c010000`.

reg-base-per-redistributor

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.
Type: `string`. Default value: `""`.

SPI-blocks

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.
Type: `int`. Default value: `0x1e`.

4.10.30 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 4-401: IP revisions support

Revision	Quality level
r0p2	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC600AE contains the following CADI targets:

- GIC600AE

GIC600AE contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3InterruptTranslationService](#)
- [GICv3ProtocolChecker](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)

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.



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 the [MSIRewriter](#) component.

MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

Table 4-402: `pvbus_s` port

PVBus attribute	Bits used	Property encoded	Notes
<code>MasterID</code>	Bits[31:0]	Master ID that invoked the register access	-
<code>ExtendedID</code>	-	-	Not used.
<code>UserFlags</code>	-	-	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 4-403: Ports

Name	Protocol	Type	Description
<code>apb_bus</code>	PVBus	Slave	APB4 port to FMU, which will be connected from safety island for FuSa
<code>chip_id</code>	Value	Slave	<code>chip_id</code> port used for multichip operation
<code>cpu_active_s[256]</code>	Signal	Slave	CPUActive pins.
<code>fmu_error_int</code>	Signal	Master	FuSa FMU error interrupt signal
<code>fmu_fault_int</code>	Signal	Master	FuSa FMU fault interrupt signal
<code>po_reset</code>	Signal	Slave	Resets. This is used as a <code>dbg_reset</code> in TRM, and also be used for cold reset for PMU and FMU.
<code>ppi_in_<n>[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu <n>, where n is in the range 0-255.
<code>pvbus_m</code>	PVBus	Master	Memory bus out: transactions generated by the IRI.
<code>pvbus_s</code>	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
<code>redistributor_m[256]</code>	GICv3Comms	Master	Input from and output to CPU interface.
<code>reset</code>	Signal	Slave	Resets.
<code>spi_in[960]</code>	Signal	Slave	Shared peripheral interrupts.
<code>wake_request[256]</code>	Signal	Master	Power management outputs.

Parameters for GIC600AE

affinity-width

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`. Default value: "4.8.8.8".

chip-id

Chip ID when multichip operation is enabled.

Type: `int`. Default value: 0x0.

chip-select-affinity-level

Affinity level 2 or 3 can be used for chip select.

Type: `int`. Default value: 0x3.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`. Default value: 0x0.

common-vPE-table-affinity

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.

Type: `string`. Default value: "".

CPU-affinities

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.

Type: `string`. Default value: "".

CPU-affinities-file

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.

Type: `string`. Default value: "".

direct-lpi-support

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: 0x0.

DS-behaviour

GICD_CTLR.DS field behaviour: 0:RAZ/WI, 1:RAO/WI, 2:RW.

Type: `int`. Default value: 0x2.

enabled

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: 0x1.

gict-allow-ns-reset

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.

Type: `bool`. Default value: 0x1.

IIDR

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.

Type: `int`. Default value: `0x0`.

ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x8`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-ID-bits

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `int`. Default value: `0x0`.

max-cores-supported-by-GCI

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`. Default value: `0x8`.

max-pe-on-chip

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if `enable-multichip-operation` is set.

Type: `bool`. Default value: `0x1`.

PPI-count

Selects the number of PPI available for each PE as 8(id22-27,29,30), 12(id 20-31) or 16(id 16-31).

Type: `int`. Default value: `0x10`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

RAS-CFI-support

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-FI-support

If true, fault handling interrupt is supported. Not supported otherwise.
Type: `bool`. Default value: `0x0`.

RAS-UE-support

If true, In-band uncorrected error reporting is supported. Not supported otherwise.
Type: `bool`. Default value: `0x0`.

RAS-UI-support

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.
Type: `bool`. Default value: `0x0`.

redistributor-group

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.
Type: `string`. Default value: `""`.

redistributor-group-file

File path to redistributor grouping information with affinity as JSON. The file uses the same format as "redistributor-group" parameter.
Type: `string`. Default value: `""`.

redistributor-power-managed-by-pwrr

GIC600 dedistributor power management is done by updating GICR_PWRR register.
Type: `bool`. Default value: `0x1`.

reg-base

GIC-600 base address.
Type: `int`. Default value: `0x2c010000`.

reg-base-per-redistributor

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.
Type: `string`. Default value: `""`.

SPI-blocks

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.
Type: `int`. Default value: `0x1e`.

4.10.31 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 4-404: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC600AE_Filter contains the following CADI targets:

- GIC600AE

GIC600AE_Filter contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3InterruptTranslationService](#)
- [GICv3ProtocolChecker](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)

About GIC600AE_Filter

GIC600AE and GIC600AE_Filter are minimal models of an Arm® GIC-600AE Generic Interrupt Controller, suitable for single-chip or multichip systems, where multichip operation has been tested, with the following exceptions:

- LPI/LPI command is implemented but untested.
- Power operation is unimplemented.

They provide a simple configuration interface that allows designers to introduce GIC600AE-like functionality to their systems, while only implementing the architectural behavior, as defined by the GICv3 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC600AE and GIC600AE_Filter have a `pvbuss_s` port for register accesses and a `pvbuss_m` port for the LPI-related traffic from redistributors and the ITS. The `pvbuss_m` port is also used to compose multichip operation.

In addition, the GIC600AE_Filter variant has a `pvbuss_filtermiss_m` port, to which any transactions on the `pvbuss_s` port and not directed to a 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 the [MSIRewriter](#) component.

MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

Table 4-405: `pvbus_s` port

PVBus attribute	Bits used	Property encoded	Notes
<code>MasterID</code>	Bits[31:0]	Master ID that invoked the register access	-
<code>ExtendedID</code>	-	-	Not used.
<code>UserFlags</code>	-	-	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 4-406: 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_<n>[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu <n>, where n is in the range 0-255.
pvbus_filtermiss_m	PVBus	Master	Memory bus out. Transactions not filtered by the component.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Reset.
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
wake_request[256]	Signal	Master	Power management outputs.

Parameters for GIC600AE

affinity-width

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: string. Default value: "4.8.8.8".

chip-id

Chip ID when multichip operation is enabled.

Type: int. Default value: 0x0.

chip-select-affinity-level

Affinity level 2 or 3 can be used for chip select.

Type: int. Default value: 0x3.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: bool. Default value: 0x0.

common-vPE-table-affinity

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.

Type: string. Default value: "".

CPU-affinities

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.

Type: `string`. Default value: `""`.

CPU-affinities-file

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.

Type: `string`. Default value: `""`.

direct-lpi-support

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: `0x0`.

DS-behaviour

GICD_CTLR.DS field behaviour: 0:RAZ/WI, 1:RAO/WI, 2:RW.

Type: `int`. Default value: `0x2`.

enabled

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

gict-allow-ns-reset

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.

Type: `bool`. Default value: `0x1`.

IIDR

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.

Type: `int`. Default value: `0x0`.

ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x8`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-ID-bits

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

Type: `int`. Default value: `0x0`.

max-cores-supported-by-GCI

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`. Default value: `0x8`.

max-pe-on-chip

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if `enable-multichip-operation` is set.

Type: `bool`. Default value: `0x1`.

PPI-count

Selects the number of PPI available for each PE as 8(id22-27,29,30), 12(id 20-31) or 16(id 16-31).

Type: `int`. Default value: `0x10`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

RAS-CFI-support

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-FI-support

If true, fault handling interrupt is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-UE-support

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-UI-support

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

redistributor-group

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.

Type: `string`. Default value: `""`.

redistributor-group-file

File path to redistributor grouping information with affinity as JSON. The file uses the same format as "redistributor-group" parameter.

Type: `string`. Default value: `""`.

redistributor-power-managed-by-pwrr

GIC600 redistributor power management is done by updating GICR_PWRR register.

Type: `bool`. Default value: `0x1`.

reg-base

GIC-600 base address.
Type: `int`. Default value: `0x2c010000`.

reg-base-per-redistributor

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.
Type: `string`. Default value: `""`.

SPI-blocks

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.
Type: `int`. Default value: `0x1e`.

4.10.32 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 4-407: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC600_Filter contains the following CADI targets:

- GIC600

GIC600_Filter contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3InterruptTranslationService](#)
- [GICv3ProtocolChecker](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)

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+.



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 the [MSIRewriter](#) component.

MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

Table 4-408: `pvbus_s` port

PVBus attribute	Bits used	Property encoded	Notes
<code>MasterID</code>	Bits[31:0]	Master ID that invoked the register access	-
<code>ExtendedID</code>	-	-	Not used.
<code>UserFlags</code>	-	-	Not used.



Note

The `pvbus_m` port does not use `MasterID`, `ExtendedID`, or `UserFlags`.

Ports for GIC600_Filter

Table 4-409: Ports

Name	Protocol	Type	Description
chip_id	Value	Slave	chip_id port which is valid from GIC600 r1p2. Writing to this port for prior GIC600 version will be ignored.
cpu_active_s[256]	Signal	Slave	CPUActive pins.
po_reset	Signal	Slave	Reset.
ppi_in_<n>[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu <n>, where n is in the range 0-255.
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

affinity-width

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: string. Default value: "4.8.8.8".

chip-id

Chip ID when multichip operation is enabled.

Type: int. Default value: 0x0.

chip-select-affinity-level

Affinity level 2 or 3 can be used for chip select.

Type: int. Default value: 0x3.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: bool. Default value: 0x0.

common-vPE-table-affinity

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.

Type: string. Default value: "".

CPU-affinities

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.

Type: string. Default value: "".

CPU-affinities-file

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.

Type: `string`. Default value: `""`.

direct-lpi-support

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: `0x0`.

DS-behaviour

GICD_CTLR.DS field behaviour: 0:RAZ/WI, 1:RAO/WI, 2:RW.

Type: `int`. Default value: `0x2`.

enabled

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

gict-allow-ns-reset

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.

Type: `bool`. Default value: `0x1`.

IIDR

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.

Type: `int`. Default value: `0x0`.

ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x8`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-ID-bits

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

Type: `int`. Default value: `0x0`.

max-cores-supported-by-GCI

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`. Default value: `0x8`.

max-pe-on-chip

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if `enable-multichip-operation` is set.

Type: `bool`. Default value: `0x1`.

PPI-count

Selects the number of PPI available for each PE as 8(id22-27,29,30), 12(id 20-31) or 16(id 16-31).

Type: `int`. Default value: `0x10`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

RAS-CFI-support

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-FI-support

If true, fault handling interrupt is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-UE-support

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-UI-support

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

redistributor-group

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.

Type: `string`. Default value: `""`.

redistributor-group-file

File path to redistributor grouping information with affinity as JSON. The file uses the same format as "redistributor-group" parameter.

Type: `string`. Default value: `""`.

redistributor-power-managed-by-pwrr

GIC600 dedistributor power management is done by updating GICR_PWRR register.

Type: `bool`. Default value: `0x1`.

reg-base

GIC-600 base address.

Type: `int`. Default value: `0x2c010000`.

reg-base-per-redistributor

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.
Type: `string`. Default value: `""`.

SPI-blocks

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.
Type: `int`. Default value: `0x1e`.

4.10.33 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 4-410: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC625 contains the following CADI targets:

- GIC625

GIC625 contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3ProtocolChecker](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)

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 and GIC625_Filter support 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

Table 4-411: 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_<n>[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu <n>, where n is in the range 0-255.
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[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

affinity-width

A dotted quad indicating the bitwidth of fields at each affinity level.
Type: string. Default value: "4.8.8.8".

chip-id

Chip ID when multichip operation is enabled.
Type: int. Default value: 0x0.

chip-select-affinity-level

Affinity level 2 or 3 can be used for chip select.
Type: int. Default value: 0x3.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`. Default value: `0x0`.

common-vPE-table-affinity

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.

Type: `string`. Default value: `""`.

CPU-affinities

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.

Type: `string`. Default value: `""`.

CPU-affinities-file

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.

Type: `string`. Default value: `""`.

direct-lpi-support

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: `0x0`.

DS-behaviour

GICD_CTLR.DS field behaviour: 0:RAZ/WI, 1:RAO/WI, 2:RW.

Type: `int`. Default value: `0x2`.

enabled

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

gict-allow-ns-reset

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.

Type: `bool`. Default value: `0x1`.

IIDR

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.

Type: `int`. Default value: `0x0`.

ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x8`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-ID-bits

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `int`. Default value: `0x0`.

max-cores-supported-by-GCI

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`. Default value: `0x8`.

max-pe-on-chip

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if `enable-multichip-operation` is set.

Type: `bool`. Default value: `0x1`.

PPI-count

Selects the number of PPI available for each PE as 8(id22-27,29,30), 12(id 20-31) or 16(id 16-31).

Type: `int`. Default value: `0x10`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

RAS-CFI-support

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-FI-support

If true, fault handling interrupt is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-UE-support

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-UI-support

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

redistributor-group

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.

Type: `string`. Default value: `""`.

redistributor-group-file

File path to redistributor grouping information with affinity as JSON. The file uses the same format as "redistributor-group" parameter.

Type: `string`. Default value: `""`.

redistributor-power-managed-by-pwrr

GIC600 dedistributor power management is done by updating GICR_PWRR register.

Type: `bool`. Default value: `0x1`.

reg-base

GIC-600 base address.

Type: `int`. Default value: `0x2c010000`.

reg-base-per-redistributor

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.

Type: `string`. Default value: `""`.

SPI-blocks

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

Type: `int`. Default value: `0x1e`.

4.10.34 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 4-412: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC625_Filter contains the following CADI targets:

- GIC625

GIC625_Filter contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3ProtocolChecker](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)

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 and GIC625_Filter support 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 4-413: 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_<n>[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu <n>, where n is in the range 0-255.
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.

Name	Protocol	Type	Description
<code>spi_in[960]</code>	Signal	Slave	Shared peripheral interrupts.
<code>wake_request[8]</code>	Signal	Master	Power management outputs.

Parameters for GIC625

affinity-width

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`. Default value: "4.8.8.8".

chip-id

Chip ID when multichip operation is enabled.

Type: `int`. Default value: `0x0`.

chip-select-affinity-level

Affinity level 2 or 3 can be used for chip select.

Type: `int`. Default value: `0x3`.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`. Default value: `0x0`.

common-vPE-table-affinity

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.

Type: `string`. Default value: "".

CPU-affinities

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.

Type: `string`. Default value: "".

CPU-affinities-file

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.

Type: `string`. Default value: "".

direct-lpi-support

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: `0x0`.

DS-behaviour

GICD_CTLR.DS field behaviour: 0:RAZ/WI, 1:RAO/WI, 2:RW.

Type: `int`. Default value: `0x2`.

enabled

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

gict-allow-ns-reset

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.

Type: `bool`. Default value: `0x1`.

IIDR

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.

Type: `int`. Default value: `0x0`.

ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x8`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-ID-bits

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

Type: `int`. Default value: `0x0`.

max-cores-supported-by-GCI

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`. Default value: `0x8`.

max-pe-on-chip

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if enable-multichip-operation is set.

Type: `bool`. Default value: `0x1`.

PPI-count

Selects the number of PPI available for each PE as 8(id22-27,29,30), 12(id 20-31) or 16(id 16-31).

Type: `int`. Default value: `0x10`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

RAS-CFI-support

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-FI-support

If true, fault handling interrupt is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-UE-support

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

RAS-UI-support

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

Type: `bool`. Default value: `0x0`.

redistributor-group

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.

Type: `string`. Default value: `""`.

redistributor-group-file

File path to redistributor grouping information with affinity as JSON. The file uses the same format as "redistributor-group" parameter.

Type: `string`. Default value: `""`.

redistributor-power-managed-by-pwrr

GIC600 dedistributor power management is done by updating GICR_PWRR register.

Type: `bool`. Default value: `0x1`.

reg-base

GIC-600 base address.

Type: `int`. Default value: `0x2c010000`.

reg-base-per-redistributor

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.

Type: `string`. Default value: `""`.

SPI-blocks

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

Type: `int`. Default value: `0x1e`.

4.10.35 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 4-414: IP revisions support

Revision	Quality level
r0p0	Full support
r1p0	Full support
r2p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC700 contains the following CADI targets:

- GIC700

GIC700 contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3ProtocolChecker](#)
- [GICv3Redistributor](#)
- [GICv4InterruptTranslationService](#)
- [GICv4RedistributorInternal](#)
- [PVBusSlave](#)

Changes in 11.24.4

Parameters added:

- `cross-chip-AMBA-is-ACE`

About GIC700

GIC700 and GIC700_Filter are minimal models of an Arm® GIC-700 Generic Interrupt Controller, suitable for single-chip or multichip systems. They support the following multichip operations:

- Multichip configuration through distributor registers.
- SPI.
- SGI.
- Physical LPI.
- Physical LPI command.
- Virtual LPI.

- Virtual LPI command.

The models provide a simple configuration interface that allows designers to introduce GIC700-like functionality to systems, while only implementing the architectural behavior, as defined by the GICv3/GICv4 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC700 and GIC700_Filter have a `pvbuss_s` port for register accesses and a `pvbuss_m` port for the LPI-related traffic from redistributors and the ITS.

In addition, the GIC700_Filter variant has a `pvbuss_filtermiss_m` port, to which any transactions on the `pvbuss_s` port and not directed to a 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.



- 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 the [MSIRewriter](#) component.

Limitations

- The following multichip operations are not yet supported:
 - vPE control.
 - Power control.
- There is no support for the `vmov` command.
- Limited testing has been performed for multichip LPI operations, so the model might contain defects.

- One ITS is supported for each GIC for multichip operations.

MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

Table 4-415: pvb_{us}_s port

PVBus attribute	Bits used	Property encoded	Notes
MasterID	Bits[31:0]	Master ID that invoked the register access	-
ExtendedID	-	-	Not used.
UserFlags	-	-	Not used.



The pvb_{us}_m port does not use `MasterID`, `ExtendedID`, or `UserFlags`.

Ports for GIC700

Table 4-416: 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<n>[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu <n>, where <n> is in the range 0-255.
extended_spi_in[1024]	Signal	Slave	Extended Shared peripheral interrupts.
icdrdt_out	PVBus	Master	AXI Stream Interface. This is used only when local cross chip addressing is enabled at the moment
icrdt_in	PVBus	Slave	-
po_reset	Signal	Slave	Resets.
ppi_in<n>[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu <n>, where n is in the range 0-255.
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

add-output-cpu-wake-request-signal-from-redistributor

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.

Type: `bool`. Default value: `0x0`.

affinity-width

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`. Default value: `"4.8.8.8"`.

allow-LPIEN-clear

Allow RW behaviour on `GICR_CTLR.LPIEN` instead of set once.

Type: `bool`. Default value: `0x1`.

chip-count

The total number of chips supported.

Type: `int`. Default value: `0x10`.

chip-id

Chip ID when multichip operation is enabled.

Type: `int`. Default value: `0x0`.

chip-select-affinity-level

Affinity level 2 or 3 can be used for chip select.

Type: `int`. Default value: `0x3`.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the `ISPENDR` register.

Type: `bool`. Default value: `0x0`.

common-vPE-table-affinity

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.

Type: `string`. Default value: `""`.

consolidators

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.

Type: `string`. Default value: `""`.

CPU-affinities

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.

Type: `string`. Default value: `""`.

CPU-affinities-file

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.

Type: `string`. Default value: `""`.

cross-chip-AMBA-is-ACE

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.

Type: `bool`. Default value: `0x0`.

direct-lpi-support

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: `0x0`.

DS-behaviour

GICD_CTLR.DS field behaviour: 0:RAZ/WI, 1:RAO/WI, 2:RW.

Type: `int`. Default value: `0x2`.

enable-local-cross-chip-addressing

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.

Type: `bool`. Default value: `0x0`.

enabled

Enable GICv3 functionality; when false the component is inactive. This will be replaced by "has_gicv3" when GICv3 is replaced by GIC.

Type: `bool`. Default value: `0x1`.

extended-ppi-count

Number of extended PPI supported.

Type: `int`. Default value: `0x40`.

extended-spi-count

Number of extended SPI supported.

Type: `int`. Default value: `0x400`.

GICD_CTLR-DS-1-means-secure-only

If GICD_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: `bool`. Default value: `0x0`.

GICD_TYPER2

GICD_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

Type: `int`. Default value: `0x0`.

GITS_BASER0-entry-bytes

Number of bytes required per entry for GITS_BASER0 register.

Type: `int`. Default value: `0x8`.

GITS_BASER0-indirect-RAZ

Indirect field for GITS_BASER0 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER0-type

Type field for GITS_BASER0 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x1`.

GITS_BASER1-entry-bytes

Number of bytes required per entry for GITS_BASER1 register.

Type: `int`. Default value: `0x2`.

GITS_BASER1-indirect-RAZ

Indirect field for GITS_BASER1 register is RAZ/WI.

Type: `bool`. Default value: `0x1`.

GITS_BASER1-type

Type field for GITS_BASER1 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x4`.

GITS_BASER2-entry-bytes

Number of bytes required per entry for GITS_BASER2 register.

Type: `int`. Default value: `0x8`.

GITS_BASER2-indirect-RAZ

Indirect field for GITS_BASER2 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER2-type

Type field for GITS_BASER2 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x2`.

has-gicv4.1

Enable GICv4.1 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

has-two-security-states

If true, has two security states.

Type: `bool`. Default value: `0x1`.

has_mpam

Enable MPAM support on ITS and RDs.

Type: `bool`. Default value: `0x0`.

IIDR

GICD_IIDR and GICR_IIDR value.

Type: `int`. Default value: `0x0`.

ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x8`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-hardware-collection-count

Number of hardware collections held exclusively in the ITS.

Type: `int`. Default value: `0x0`.

ITS-ID-bits

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `int`. Default value: `0x0`.

ITS-vmovp-bit

Device supports software issuing a VMOVOP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVOP command across all ITSs that have mapping for that vPE.

Type: `bool`. Default value: `0x0`.

max-cores-supported-by-GCI

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`. Default value: `0x8`.

max-pe-on-chip

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if `enable-multichip-operation` is set.

Type: `bool`. Default value: `0x1`.

PPI-count

Selects the number of PPI available for each PE as 8(id22-27,29,30), 12(id 20-31) or 16(id 16-31).

Type: `int`. Default value: `0x10`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

prog-mpidr

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).

Type: `int`. Default value: `0x0`.

redistributor-group

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.

Type: `string`. Default value: `""`.

redistributor-group-file

File path to redistributor grouping information with affinity as JSON. The file uses the same format as "redistributor-group" parameter.

Type: `string`. Default value: `""`.

redistributor-power-managed-by-pwrr

GIC-700 redistributor power management is done by updating GICR_PWRR register.

Type: `bool`. Default value: `0x1`.

reg-base

GIC-700 base address.

Type: `int`. Default value: `0x2c010000`.

SPI-blocks

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

Type: `int`. Default value: `0x1e`.

4.10.36 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 4-417: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC700_Filter contains the following CADI targets:

- GIC700

GIC700_Filter contains the following MTI components:

- [GICv3Distributor](#)

- [GICv3IRI](#)
- [GICv3ProtocolChecker](#)
- [GICv3Redistributor](#)
- [GICv4InterruptTranslationService](#)
- [GICv4RedistributorInternal](#)
- [PVBUSSlave](#)

Changes in 11.24.4

Parameters added:

- `cross-chip-AMBA-is-ACE`

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 `pvbuss_s` port for register accesses and a `pvbuss_m` port for the LPI-related traffic from redistributors and the ITS.

In addition, the GIC700_Filter variant has a `pvbuss_filtermiss_m` port, to which any transactions on the `pvbuss_s` port and not directed to a 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.



- 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 the [MSIRewriter](#) component.

Limitations

- The following multichip operations are not yet supported:
 - vPE control.
 - Power control.
- There is no support for the `vmov` command.
- Limited testing has been performed for multichip LPI operations, so the model might contain defects.
- One ITS is supported for each GIC for multichip operations.

MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

Table 4-418: pvbuss port

PVBus attribute	Bits used	Property encoded	Notes
MasterID	Bits[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 4-419: Ports

Name	Protocol	Type	Description
axi_stream_msi_s[32]	PVBus	Slave	AXI Stream Interface ports for MSI (message-signalled interrupt) translation, in GIC-700. Typically the SMMU's TCU connects to this port for MSI.
chip_id	Value	Slave	chip_id port for multichip operation
cpu_active_s[256]	Signal	Slave	CPUActive pins.
extended_ppi_in_<n>[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu <n>, where <n> is in the range 0-255.
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_<n>[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu <n>, where n is in the range 0-255.
pvbust_filtermiss_m	PVBus	Master	Memory bus out. Transactions not filtered by the component.
pvbust_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbust_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

add-output-cpu-wake-request-signal-from-redistributor

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.
Type: bool. Default value: 0x0.

affinity-width

A dotted quad indicating the bitwidth of fields at each affinity level.
Type: string. Default value: "4.8.8.8".

allow-LPIEN-clear

Allow RW behaviour on GICR_CTLR.LPIEN instead of set once.
Type: bool. Default value: 0x1.

chip-count

The total number of chips supported.
Type: int. Default value: 0x10.

chip-id

Chip ID when multichip operation is enabled.
Type: int. Default value: 0x0.

chip-select-affinity-level

Affinity level 2 or 3 can be used for chip select.

Type: `int`. Default value: `0x3`.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`. Default value: `0x0`.

common-vPE-table-affinity

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.

Type: `string`. Default value: `""`.

consolidators

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.

Type: `string`. Default value: `""`.

CPU-affinities

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.

Type: `string`. Default value: `""`.

CPU-affinities-file

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.

Type: `string`. Default value: `""`.

cross-chip-AMBA-is-ACE

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.

Type: `bool`. Default value: `0x0`.

direct-lpi-support

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: `0x0`.

DS-behaviour

GICD_CTLR.DS field behaviour: 0:RAZ/WI, 1:RAO/WI, 2:RW.

Type: `int`. Default value: `0x2`.

enable-local-cross-chip-addressing

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.

Type: `bool`. Default value: `0x0`.

enabled

Enable GICv3 functionality; when false the component is inactive. This will be replaced by "has_gicv3" when GICv3 is replaced by GIC.

Type: `bool`. Default value: `0x1`.

extended-ppi-count

Number of extended PPI supported.

Type: `int`. Default value: `0x40`.

extended-spi-count

Number of extended SPI supported.

Type: `int`. Default value: `0x400`.

GICD_CTLR_DS-1-means-secure-only

If GICD_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: `bool`. Default value: `0x0`.

GICD_TYPER2

GICD_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

Type: `int`. Default value: `0x0`.

GITS_BASER0-entry-bytes

Number of bytes required per entry for GITS_BASER0 register.

Type: `int`. Default value: `0x8`.

GITS_BASER0-indirect-RAZ

Indirect field for GITS_BASER0 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER0-type

Type field for GITS_BASER0 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x1`.

GITS_BASER1-entry-bytes

Number of bytes required per entry for GITS_BASER1 register.

Type: `int`. Default value: `0x2`.

GITS_BASER1-indirect-RAZ

Indirect field for GITS_BASER1 register is RAZ/WI.

Type: `bool`. Default value: `0x1`.

GITS_BASER1-type

Type field for GITS_BASER1 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x4`.

GITS_BASER2-entry-bytes

Number of bytes required per entry for GITS_BASER2 register.

Type: `int`. Default value: `0x8`.

GITS_BASER2-indirect-RAZ

Indirect field for GITS_BASER2 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER2-type

Type field for GITS_BASER2 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x2`.

has-gicv4.1

Enable GICv4.1 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

has-two-security-states

If true, has two security states.

Type: `bool`. Default value: `0x1`.

has_mpam

Enable MPAM support on ITS and RDs.

Type: `bool`. Default value: `0x0`.

IIDR

GICD_IIDR and GICR_IIDR value.

Type: `int`. Default value: `0x0`.

ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x8`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-hardware-collection-count

Number of hardware collections held exclusively in the ITS.

Type: `int`. Default value: `0x0`.

ITS-ID-bits

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `int`. Default value: `0x0`.

ITS-vmovp-bit

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.

Type: `bool`. Default value: `0x0`.

max-cores-supported-by-GCI

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`. Default value: `0x8`.

max-pe-on-chip

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if `enable-multichip-operation` is set.

Type: `bool`. Default value: `0x1`.

PPI-count

Selects the number of PPI available for each PE as 8(id22-27,29,30), 12(id 20-31) or 16(id 16-31).

Type: `int`. Default value: `0x10`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

prog-mpidr

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).

Type: `int`. Default value: `0x0`.

redistributor-group

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.

Type: `string`. Default value: `""`.

redistributor-group-file

File path to redistributor grouping information with affinity as JSON. The file uses the same format as "redistributor-group" parameter.

Type: `string`. Default value: `""`.

redistributor-power-managed-by-pwrr

GIC-700 redistributor power management is done by updating GICR_PWRR register.

Type: `bool`. Default value: `0x1`.

reg-base

GIC-700 base address.
Type: `int`. Default value: `0x2c010000`.

SPI-blocks

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.
Type: `int`. Default value: `0x1e`.

4.10.37 GIC_400

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-420: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC_400 contains the following CADI targets:

- GIC_400

GIC_400 contains the following MTI components:

- [GIC_400](#)
- [PVBusSlave](#)

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 distinguish from which processor a transaction originates. 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 cluster.

CLUSTERID is the 4-bit field that either a parameter on the processor sets or a value on 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.

In using `inout_port_number_to_use`, the GIC-400 has some input and output ports that pair with a particular processor interface. For example:

- The `irqcpu[]` pin wires to the `irq` port of the corresponding processor.
- The `cntpnsirq` pin from the processor wires to a `cntpnsirq[]` pin on GIC-400 to transport a *Private Peripheral Interrupt* (PPI) from the processor to the GIC-400.

The `interfaceN.inout_port_number_to_use` parameter supports clusters that can have variable numbers of cores. It tells the GIC-400 that to send to or receive a signal from the processor that is attached to interface N, it must use these pins:

- `irqout[interfaceN.inout_port_number_to_use]`.
- `fiqout[interfaceN.inout_port_number_to_use]`.
- `virqout[interfaceN.inout_port_number_to_use]`.
- `vfiqout[interfaceN.inout_port_number_to_use]`.
- `legacyirq[interfaceN.inout_port_number_to_use]`.
- `cntpnsirq[interfaceN.inout_port_number_to_use]`.
- `cntpsirq[interfaceN.inout_port_number_to_use]`.
- `legacyfiq[interfaceN.inout_port_number_to_use]`.
- `cntvirq[interfaceN.inout_port_number_to_use]`.
- `cnthpirq[interfaceN.inout_port_number_to_use]`.
- ...

`legacyirq` and `legacyfiq` are not signals from the processor but are signals into the GIC-400 from the legacy interrupt system. They are wired to PPIs. If the control registers of the GIC-400 are set up in particular ways, they can also bypass the GIC-400. See [ARM Generic Interrupt Controller Architecture version 2.0 Architecture Specification](#) for more information.

The fabric between the clusters and the GIC might remap the `master_id` of a transaction. If so, then the GIC might lose the ability to identify the originating processor. The fabrics that ARM ships in Fast Models perform no such transformation.

The comparison that the GIC-400 performs on the `master_id` is only on the bottom 6 bits of the `master_id`. It ignores the rest. If you are writing your own fabric and do not properly propagate the `master_id` or transform it, the GIC-400 might not be able to identify the processor. The source code for the GIC_400 component can be examined to see how it might be adapted for it to understand different `master_id` schemes.

Differences between the model and the RTL

The GIC-400 model has these limitations:

- Reads and writes to GICD_ISACTIVERn/GICD_ICACTIVERn/GICD_ISPENDRn/GICD_ICPENDRn might not work as expected unless there is a configured target in GICD_ICFGRm.
- Some of the interaction of GICD_CTLR.EnableGrpX and level sensitive interrupts might not work correctly.
- It does not model the signals nIRQOUT/nFIQOUT.
- It models interrupts with positive logic, rather than the negative logic that the hardware uses. Hence, the signal pins omit the “n” prefix in their names.

Ports for GIC_400

Table 4-421: Ports

Name	Protocol	Type	Description
<code>cfgsdisable</code>	Signal	Slave	Disable write access to some GIC registers.
<code>cnthpirq[8]</code>	Signal	Slave	Secure physical timer event. PPI interrupt id 26.
<code>cntpnsirq[8]</code>	Signal	Slave	Non-secure physical timer event. PPI interrupt id 30.
<code>cntpsirq[8]</code>	Signal	Slave	Secure physical timer event. PPI interrupt id 29.
<code>cntvirq[8]</code>	Signal	Slave	Virtual timer event. PPI interrupt id 27.
<code>fiqcpu[8]</code>	Signal	Master	FIQ signal to the corresponding processor.
<code>fiqout[8]</code>	Signal	Master	FIQOUT signal to the corresponding processor.
<code>irqcpu[8]</code>	Signal	Master	IRQ signal to the corresponding processor.
<code>irqout[8]</code>	Signal	Master	IRQOUT signal to the corresponding processor.
<code>irqs[480]</code>	Signal	Slave	Interrupt request input lines for the GIC.
<code>legacyfiq[8]</code>	Signal	Slave	Signal into the GIC-400 from the legacy interrupt system. PPI interrupt id 28.
<code>legacyirq[8]</code>	Signal	Slave	Signal into the GIC-400 from the legacy interrupt system. PPI interrupt id 31.
<code>pvbus_s</code>	PVBus	Slave	Handles incoming transactions from PVBus masters.
<code>reset_signal</code>	Signal	Slave	Reset signal input.
<code>vfiqcpu[8]</code>	Signal	Master	Virtual FIQ signal to the processor.
<code>virqcpu[8]</code>	Signal	Master	Virtual IRQ signal to the processor.

Parameters for GIC_400

`enable_log_errors`

Enable logging of errors.

Type: `bool`. Default value: `0x0`.

enable_log_fatal

Enable logging of fatal errors.

Type: `bool`. Default value: `0x0`.

enable_log_warnings

Enable logging of warnings.

Type: `bool`. Default value: `0x0`.

enabled

Enable the component. If it is disabled then all register writes will have no effect.

Type: `bool`. Default value: `0x1`.

4.10.38 GIC_IRI

GIC metacomponent for redistribution of interrupts(contains Distributor, and configurable numbers 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 4-422: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC_IRI contains the following CADI targets:

- GIC_IRI

GIC_IRI contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)

Changes in 11.24.4

Parameters added:

- `cross-chip-AMBA-is-ACE`

Ports for GIC_IRI

Table 4-423: Ports

Name	Protocol	Type	Description
<code>cfgsdisable</code>	Signal	Slave	Disable some SPIs signal.
<code>extended_ppi_in_<n>[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu <n>, where <n> is in the range 0-255.

Name	Protocol	Type	Description
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_<n>[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu <n>, where n is in the range 0-255.
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

Device supports affinity level 3 values that are non-zero.

Type: bool. Default value: 0x0.

add-output-cpu-wake-request-signal-from-redistributor

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.

Type: bool. Default value: 0x0.

allow-LPIEN-clear

Allow RW behaviour on GICR_CTLR.LPIEN instead of set once.

Type: bool. Default value: 0x0.

ARE-fixed-to-one

GICv2 compatibility is not supported and GICD_CTLR.ARE_* is always one.

Type: bool. Default value: 0x0.

chip-count

The total number of chips supported.

Type: int. Default value: 0x10.

chip-id

Chip ID when multichip operation is enabled.

Type: int. Default value: 0x0.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`. Default value: `0x0`.

common-lpi-configuration

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).

Type: `int`. Default value: `0x0`.

common-vPE-table-affinity

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.

Type: `string`. Default value: `""`.

consolidators

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.

Type: `string`. Default value: `""`.

CPU-affinities

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.

Type: `string`. Default value: `""`.

CPU-affinities-file

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.

Type: `string`. Default value: `""`.

cross-chip-AMBA-is-ACE

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.

Type: `bool`. Default value: `0x0`.

delay-ITS-accesses

Delay accesses from the ITS until GICR_SYNCRR is read.

Type: `bool`. Default value: `0x1`.

delay-redistributor-accesses

Delay memory accesses from the redistributor until GICR_SYNCRR is read.

Type: `bool`. Default value: `0x1`.

direct-lpi-support

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: `0x0`.

DPG-ARE-only

Limit application of DPG bits to interrupt groups for which ARE=1.

Type: bool. Default value: 0x0.

DPG-bits-implemented

Enable implementation of interrupt group participation bits or DPG bits in GICR_CTLR.

Type: bool. Default value: 0x0.

DS-fixed-to-zero

Enable/disable support of single security state.

Type: bool. Default value: 0x0.

enable-local-cross-chip-addressing

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.

Type: bool. Default value: 0x0.

enable-multichip-operation

Enables multi-chip operation between Distributors in distributed GIC IRI.

Type: bool. Default value: 0x0.

enable_protocol_checking

Enable/disable protocol checking at cpu interface.

Type: bool. Default value: 0x0.

enabled

Enable GICv3 functionality; when false the component is inactive.

Type: bool. Default value: 0x1.

extended-ppi-count

Number of extended PPI supported.

Type: int. Default value: 0x0.

extended-spi-count

Number of extended SPI supported.

Type: int. Default value: 0x0.

fixed-routed-spis

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.

Type: string. Default value: "".

GICD-alias

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).

Type: int. Default value: 0x0.

GICD-legacy-registers-as-reserved

When ARE is RAO/WI, makes superfluous registers in GICD reserved (including for the purpose of STATUSR updates).

Type: `bool`. Default value: `0x0`.

GICD_CTLR-DS-1-means-secure-only

If `GICD_CTLR.DS=1`, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: `bool`. Default value: `0x0`.

GICD_ITARGETSR-RAZWI

If true, the `GICD_ITARGETS` registers are RAZ/WI.

Type: `bool`. Default value: `0x0`.

GICD_PIDR

The value for the `GICD_PIDR` registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `int`. Default value: `0x0`.

GICR-clear-enable-supported

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.

Type: `bool`. Default value: `0x0`.

GICR-invalidate-registers-implemented

When true, the registers `GICR_INVLPIR`, `GICR_INVALLR` and `GICR_SYNCR` are implemented.

Type: `bool`. Default value: `0x0`.

GICR_PIDR

The value for the `GICR_PIDR` registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `int`. Default value: `0x0`.

GICR_PROPBASER-read-only

`GICR_PROPBASER` register is read-only.

Type: `bool`. Default value: `0x0`.

GICR_PROPBASER-reset-value

Value of `GICR_PROPBASER` on reset.

Type: `int`. Default value: `0x0`.

gicv2-only

If true, when using the GICv3 model, pretend to be a GICv2 system.

Type: `bool`. Default value: `0x0`.

GITS_BASER0-entry-bytes

Number of bytes required per entry for `GITS_BASER0` register.

Type: `int`. Default value: `0x8`.

GITS_BASER0-indirect-RAZ

Indirect field for `GITS_BASER0` register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER0-type

Type field for `GITS_BASER0` register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER1-entry-bytes

Number of bytes required per entry for GITS_BASER1 register.

Type: `int`. Default value: `0x8`.

GITS_BASER1-indirect-RAZ

Indirect field for GITS_BASER1 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER1-type

Type field for GITS_BASER1 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER2-entry-bytes

Number of bytes required per entry for GITS_BASER2 register.

Type: `int`. Default value: `0x8`.

GITS_BASER2-indirect-RAZ

Indirect field for GITS_BASER2 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER2-type

Type field for GITS_BASER2 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER3-entry-bytes

Number of bytes required per entry for GITS_BASER3 register.

Type: `int`. Default value: `0x8`.

GITS_BASER3-indirect-RAZ

Indirect field for GITS_BASER3 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER3-type

Type field for GITS_BASER3 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER4-entry-bytes

Number of bytes required per entry for GITS_BASER4 register.

Type: `int`. Default value: `0x8`.

GITS_BASER4-indirect-RAZ

Indirect field for GITS_BASER4 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER4-type

Type field for GITS_BASER4 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER5-entry-bytes

Number of bytes required per entry for GITS_BASER5 register.

Type: `int`. Default value: `0x8`.

GITS_BASER5-indirect-RAZ

Indirect field for GITS_BASER5 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER5-type

Type field for GITS_BASER5 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER6-entry-bytes

Number of bytes required per entry for GITS_BASER6 register.

Type: `int`. Default value: `0x8`.

GITS_BASER6-indirect-RAZ

Indirect field for GITS_BASER6 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER6-type

Type field for GITS_BASER6 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER7-entry-bytes

Number of bytes required per entry for GITS_BASER7 register.

Type: `int`. Default value: `0x8`.

GITS_BASER7-indirect-RAZ

Indirect field for GITS_BASER7 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER7-type

Type field for GITS_BASER7 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_PIDR

The value for the GITS_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `int`. Default value: `0x0`.

has-two-security-states

If true, has two security states.

Type: `bool`. Default value: `0x1`.

has_mpam

Enable MPAM support on ITS and RDs.

Type: `bool`. Default value: `0x0`.

has_VPENDBASER-dirty-flag-on-load

GICR_VPENDBASER.Dirty reflects transient loading state when valid=1.
Type: bool. Default value: 0x0.

ICFGR-PPI-mask

Mask for writes to ICFGR registers that configure PPIs.
Type: int. Default value: 0xaaaaaaaa.

ICFGR-PPI-reset

Reset value for ICFGR registers that configure PPIs.
Type: int. Default value: 0x0.

ICFGR-rsvd-bit

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGRn for n>0.
Type: bool. Default value: 0x0.

ICFGR-SGI-mask

Mask for writes to ICFGR registers that configure SGIs.
Type: int. Default value: 0x0.

ICFGR-SGI-reset

Reset value for ICFGR registers that configure SGIs.
Type: int. Default value: 0xaaaaaaaa.

ICFGR-SPI-mask

Mask for writes to ICFGR registers that configure SPIs.
Type: int. Default value: 0xaaaaaaaa.

ICFGR-SPI-reset

Reset value for ICFGR registers that configure SPIs.
Type: int. Default value: 0x0.

ignore-generate-sgi-when-no-are

Ignore GenerateSgi packets coming from the CPU interface if both ARE_S and ARE_NS are 0.
Type: bool. Default value: 0x0.

IGROUP-PPI-mask

Mask for writes to PPI bits in IGROUP registers.
Type: int. Default value: 0xfffff.

IGROUP-PPI-reset

Reset value for SGI bits in IGROUP registers.
Type: int. Default value: 0x0.

IGROUP-SGI-mask

Mask for writes to SGI bits in IGROUP registers.
Type: int. Default value: 0xfffff.

IGROUP-SGI-reset

Reset value for SGI bits in IGROUP registers.
Type: int. Default value: 0x0.

IIDR

GICD_IIDR and GICR_IIDR value.

Type: `int`. Default value: `0x0`.

IRI-ID-bits

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.

Type: `int`. Default value: `0x10`.

irouter-default-mask

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

Type: `string`. Default value: `""`.

irouter-default-reset

Default Reset Value of IROUTER[32..1019] register in the form 'a.b.c.d' or '*'.

Type: `string`. Default value: `""`.

IROUTER-IRM-RAZ-WI

GICD_IROUTERn.InterruptRoutingMode is RAZ/WI.

Type: `bool`. Default value: `0x0`.

irouter-mask-values

Mask Value of IROUTER[n] register in the form 'n=a.b.c.d'. n can be ≥ 32 and ≤ 1019 .

Type: `string`. Default value: `""`.

irouter-reset-values

Reset Value of IROUTER[n] register in the form 'n=a.b.c.d' or 'n=*'. n can be ≥ 32 and ≤ 1019 .

Type: `string`. Default value: `""`.

ITS-BASER-force-page-alignment

Force alignment of address written to a GITS_BASER register to the page size configured.

Type: `bool`. Default value: `0x1`.

ITS-cache-invalidate-on-disable

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>.

Type: `bool`. Default value: `0x0`.

ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x0`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x0`.

ITS-cumulative-collection-tables

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.

Type: `bool`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-entry-size

Number of bytes required to store each entry in the ITT tables.

Type: `int`. Default value: `0x8`.

ITS-hardware-collection-count

Number of hardware collections held exclusively in the ITS.

Type: `int`. Default value: `0x0`.

ITS-ID-bits

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

ITS-legacy-iidr-typer-offset

Put the GITS_IIDR and GITS_TYPER registers at their older offset of 0x8 and 0x4 respectively.

Type: `bool`. Default value: `0x0`.

ITS-MOVALL-update-collections

Whether MOVALL command updates the collection entires.

Type: `bool`. Default value: `0x0`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

Type: `int`. Default value: `0x0`.

ITS-threaded-command-queue

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

ITS-TRANSLATE64R

Add an implementation specific register at 0x10008 supporting 64 bit TRANSLATER (dev[63:32], interrupt[31:0]).

Type: `bool`. Default value: `0x0`.

ITS-use-physical-target-addresses

Use physical hardware addresses for targets in ITS commands -- must be true for distributed implementations.

Type: `bool`. Default value: `0x1`.

ITS-vmovp-bit

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.

Type: `bool`. Default value: `0x0`.

ITS0-base

Register base address for ITS0 (automatic if 0).

Type: `int`. Default value: `0x0`.

ITS1-base

Register base address for ITS1 (automatic if 0).

Type: `int`. Default value: `0x0`.

ITS2-base

Register base address for ITS2 (automatic if 0).

Type: `int`. Default value: `0x0`.

ITS3-base

Register base address for ITS3 (automatic if 0).

Type: `int`. Default value: `0x0`.

legacy-sgi-enable-rao

Enables for SGI associated with an ARE=0 regime are RAO/WI.

Type: `bool`. Default value: `0x0`.

local-SEIs

Generate SEI to signal internal issues.

Type: `bool`. Default value: `0x0`.

local-VSEIs

Generate VSEI to signal internal issues.

Type: `bool`. Default value: `0x0`.

lockable-SPI-count

Number of SPIs that are locked down when CFGSDISABLE signal is asserted. Only applies for GICv2.

Type: `int`. Default value: `0x0`.

LPI-cache-check-data

Enable Cached LPI data against memory checking when available for cache type.

Type: `bool`. Default value: `0x0`.

LPI-cache-type

Cache type for LPIs, 0:No caching, 1:Full caching.

Type: `int`. Default value: `0x1`.

max-pe-on-chip

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

monolithic

Indicate that the implementation is not distributed.

Type: `bool`. Default value: `0x0`.

mpam_max_partid

Maximum valid PARTID.

Type: `int`. Default value: `0xffff`.

mpam_max_pmg

Maximum valid PMG.

Type: `int`. Default value: `0xff`.

MSI_IIDR

Value returned in MSI_IIDR registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame0-base

If non-zero, sets the base address used for non-secure MSI frame 0 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame0-max-SPI

Maximum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame0-min-SPI

Minimum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame1-base

If non-zero, sets the base address used for non-secure MSI frame 1 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame1-max-SPI

Maximum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame1-min-SPI

Minimum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame2-base

If non-zero, sets the base address used for non-secure MSI frame 2 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame2-max-SPI

Maximum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame2-min-SPI

Minimum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame3-base

If non-zero, sets the base address used for non-secure MSI frame 3 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame3-max-SPI

Maximum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame3-min-SPI

Minimum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame4-base

If non-zero, sets the base address used for non-secure MSI frame 4 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame4-max-SPI

Maximum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame4-min-SPI

Minimum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame5-base

If non-zero, sets the base address used for non-secure MSI frame 5 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame5-max-SPI

Maximum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame5-min-SPI

Minimum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame6-base

If non-zero, sets the base address used for non-secure MSI frame 6 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame6-max-SPI

Maximum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame6-min-SPI

Minimum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame7-base

If non-zero, sets the base address used for non-secure MSI frame 7 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame7-max-SPI

Maximum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame7-min-SPI

Minimum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_PIDR

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.

Type: `int`. Default value: `0x0`.

MSI_S-frame0-base

If non-zero, sets the base address used for secure MSI frame 0 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame0-max-SPI

Maximum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame0-min-SPI

Minimum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame1-base

If non-zero, sets the base address used for secure MSI frame 1 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame1-max-SPI

Maximum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame1-min-SPI

Minimum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame2-base

If non-zero, sets the base address used for secure MSI frame 2 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame2-max-SPI

Maximum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame2-min-SPI

Minimum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame3-base

If non-zero, sets the base address used for secure MSI frame 3 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame3-max-SPI

Maximum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame3-min-SPI

Minimum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame4-base

If non-zero, sets the base address used for secure MSI frame 4 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame4-max-SPI

Maximum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame4-min-SPI

Minimum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame5-base

If non-zero, sets the base address used for secure MSI frame 5 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame5-max-SPI

Maximum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame5-min-SPI

Minimum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame6-base

If non-zero, sets the base address used for secure MSI frame 6 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame6-max-SPI

Maximum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame6-min-SPI

Minimum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame7-base

If non-zero, sets the base address used for secure MSI frame 7 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame7-max-SPI

Maximum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame7-min-SPI

Minimum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if `enable-multichip-operation` is set.

Type: `bool`. Default value: `0x1`.

non-ARE-core-count

Maximum number of non-ARE cores; normally used to pass the cluster-level `NUM_CORES` parameter to the top-level redistributor.

Type: `int`. Default value: `0x8`.

outer-cacheability-support

Allow configuration of outer cacheability attributes in ITS and Redistributor.

Type: `bool`. Default value: `0x0`.

output_attributes

User-defined transform to be applied to bus attributes like `MasterID`, `ExtendedID` or `UserFlags`. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`. Default value: `"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS"`.

PA_SIZE

Number of valid bits in physical address.

Type: `int`. Default value: `0x30`.

PPI-implemented-mask

Mask of PPIs that are implemented. One bit per PPI bit 0 == PPI 16 (first PPI). This will affect other masks.

Type: `int`. Default value: `0xffff`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

priority-bits

Number of implemented priority bits.

Type: `int`. Default value: `0x5`.

processor-numbers

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.

Type: `string`. Default value: `""`.

redistributor-threaded-sync

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

reg-base

Base for decoding GICv3 registers.

Type: `int`. Default value: `0x2c010000`.

reg-base-per-redistributor

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.

Type: `string`. Default value: `""`.

reg-base-per-redistributor-file

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.

Type: `string`. Default value: `""`.

report-MSI-error-via-statusr

Report MSI error via GITS_STATUSR. (0:unsupported, 1:report by GITS_STATUSR, 2:report by GITS_STATUSR and interrupt as well).

Type: `int`. Default value: `0x0`.

sgi-range-selector-support

Device has support for the Range Selector feature for SGI.

Type: `bool`. Default value: `0x0`.

single-set-support

When true, forces redistributors to recall interrupts with a clear rather than issue a second Set command.

Type: `bool`. Default value: `0x0`.

SPI-count

Number of SPIs that are implemented.

Type: `int`. Default value: `0xe0`.

SPI-message-based-support

Distributor supports message based signaling of SPI.

Type: `bool`. Default value: `0x1`.

SPI-unimplemented

A comma separated list of unimplemented SPIs ranges for sparse SPI definition(for ex: '35, 39-42, 73').

Type: `string`. Default value: `""`.

STATUSR-implemented

Determines whether the GICR_STATUSR register is implemented.

Type: `bool`. Default value: `0x1`.

supports-shareability

Device supports shareability attributes on outgoing memory bus (i.e. is modelling an ACElite port rather than an AXI4 port).

Type: `bool`. Default value: `0x1`.

- trace-speculative-lpi-property-update**
Trace LPI property updates performed on speculative accesses (useful for debugging LPI).
Type: `bool`. Default value: `0x0`.
- virtual-lpi-support**
GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.
Type: `bool`. Default value: `0x0`.
- virtual-priority-bits**
Number of implemented virtual priority bits.
Type: `int`. Default value: `0x5`.
- wakeup-on-reset**
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.
Type: `bool`. Default value: `0x0`.

4.10.39 GIC_IRI_Filter

GIC metacomponent for redistribution of interrupts(contains Distributor, and configurable numbers 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 4-424: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GIC_IRI_Filter contains the following CADI targets:

- `GIC_IRI`

GIC_IRI_Filter contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)

Changes in 11.24.4

Parameters added:

- `cross-chip-AMBA-is-ACE`

Ports for GIC_IRI_Filter

Table 4-425: Ports

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	Disable some SPIs signal.
extended_ppi_in_<n>[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu <n>, where <n> is in the range 0-255.
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_<n>[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu <n>, where n is in the range 0-255.
pvbus_filtermiss_m	PVBus	Master	Passthrough for accesses to pages not used by the GIC IRI.
pvbus_m	PVBus	Master	Memory bus for transactions generated by the GIC.
pvbus_s	PVBus	Slave	Memory bus in.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Resets.
spi_in[988]	Signal	Slave	Shared peripheral interrupts.
wake_request[256]	Signal	Master	Power management outputs.
wire_to_msi_in_0[1024]	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 0.
wire_to_msi_in_1[1024]	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 1.
wire_to_msi_in_2[1024]	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 2.
wire_to_msi_in_3[1024]	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 3.

Parameters for GIC_IRI

A3-affinity-supported

Device supports affinity level 3 values that are non-zero.

Type: bool. Default value: 0x0.

add-output-cpu-wake-request-signal-from-redistributor

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.

Type: bool. Default value: 0x0.

allow-LPIEN-clear

Allow RW behaviour on GICR_CTLR.LPIEN instead of set once.

Type: bool. Default value: 0x0.

ARE-fixed-to-one

GICv2 compatibility is not supported and GICD_CTLR.ARE_* is always one.

Type: bool. Default value: 0x0.

chip-count

The total number of chips supported.

Type: `int`. Default value: `0x10`.

chip-id

Chip ID when multichip operation is enabled.

Type: `int`. Default value: `0x0`.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`. Default value: `0x0`.

common-lpi-configuration

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.

Type: `int`. Default value: `0x0`.

common-vPE-table-affinity

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.

Type: `string`. Default value: `""`.

consolidators

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.

Type: `string`. Default value: `""`.

CPU-affinities

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.

Type: `string`. Default value: `""`.

CPU-affinities-file

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.

Type: `string`. Default value: `""`.

cross-chip-AMBA-is-ACE

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.

Type: `bool`. Default value: `0x0`.

delay-ITS-accesses

Delay accesses from the ITS until GICR_SYNC is read.

Type: `bool`. Default value: `0x1`.

delay-redistributor-accesses

Delay memory accesses from the redistributor until GICR_SYNCR is read.

Type: `bool`. Default value: `0x1`.

direct-lpi-support

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: `0x0`.

DPG-ARE-only

Limit application of DPG bits to interrupt groups for which ARE=1.

Type: `bool`. Default value: `0x0`.

DPG-bits-implemented

Enable implementation of interrupt group participation bits or DPG bits in GICR_CTLR.

Type: `bool`. Default value: `0x0`.

DS-fixed-to-zero

Enable/disable support of single security state.

Type: `bool`. Default value: `0x0`.

enable-local-cross-chip-addressing

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.

Type: `bool`. Default value: `0x0`.

enable-multichip-operation

Enables multi-chip operation between Distributors in distributed GIC IRI.

Type: `bool`. Default value: `0x0`.

enable_protocol_checking

Enable/disable protocol checking at cpu interface.

Type: `bool`. Default value: `0x0`.

enabled

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

extended-ppi-count

Number of extended PPI supported.

Type: `int`. Default value: `0x0`.

extended-spi-count

Number of extended SPI supported.

Type: `int`. Default value: `0x0`.

fixed-routed-spis

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 .

Type: `string`. Default value: `""`.

GICD-alias

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).

Type: `int`. Default value: `0x0`.

GICD-legacy-registers-as-reserved

When ARE is RAO/WI, makes superfluous registers in GICD reserved (including for the purpose of STATUSR updates).

Type: `bool`. Default value: `0x0`.

GICD_CTLR-DS-1-means-secure-only

If GICD_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: `bool`. Default value: `0x0`.

GICD_ITARGETSR-RAZWI

If true, the GICD_ITARGETS registers are RAZ/WI.

Type: `bool`. Default value: `0x0`.

GICD_PIDR

The value for the GICD_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `int`. Default value: `0x0`.

GICR-clear-enable-supported

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.

Type: `bool`. Default value: `0x0`.

GICR-invalidate-registers-implemented

When true, the registers GICR_INVLPIR, GICR_INVALLR and GICR_SYNCRR are implemented.

Type: `bool`. Default value: `0x0`.

GICR_PIDR

The value for the GICR_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `int`. Default value: `0x0`.

GICR_PROPBASER-read-only

GICR_PROPBASER register is read-only.

Type: `bool`. Default value: `0x0`.

GICR_PROPBASER-reset-value

Value of GICR_PROPBASER on reset.

Type: `int`. Default value: `0x0`.

gicv2-only

If true, when using the GICv3 model, pretend to be a GICv2 system.

Type: `bool`. Default value: `0x0`.

GITS_BASER0-entry-bytes

Number of bytes required per entry for GITS_BASER0 register.

Type: `int`. Default value: `0x8`.

GITS_BASER0-indirect-RAZ

Indirect field for GITS_BASER0 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER0-type

Type field for GITS_BASER0 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER1-entry-bytes

Number of bytes required per entry for GITS_BASER1 register.

Type: `int`. Default value: `0x8`.

GITS_BASER1-indirect-RAZ

Indirect field for GITS_BASER1 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER1-type

Type field for GITS_BASER1 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER2-entry-bytes

Number of bytes required per entry for GITS_BASER2 register.

Type: `int`. Default value: `0x8`.

GITS_BASER2-indirect-RAZ

Indirect field for GITS_BASER2 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER2-type

Type field for GITS_BASER2 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER3-entry-bytes

Number of bytes required per entry for GITS_BASER3 register.

Type: `int`. Default value: `0x8`.

GITS_BASER3-indirect-RAZ

Indirect field for GITS_BASER3 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER3-type

Type field for GITS_BASER3 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER4-entry-bytes

Number of bytes required per entry for GITS_BASER4 register.

Type: `int`. Default value: `0x8`.

GITS_BASER4-indirect-RAZ

Indirect field for GITS_BASER4 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER4-type

Type field for GITS_BASER4 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER5-entry-bytes

Number of bytes required per entry for GITS_BASER5 register.

Type: `int`. Default value: `0x8`.

GITS_BASER5-indirect-RAZ

Indirect field for GITS_BASER5 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER5-type

Type field for GITS_BASER5 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER6-entry-bytes

Number of bytes required per entry for GITS_BASER6 register.

Type: `int`. Default value: `0x8`.

GITS_BASER6-indirect-RAZ

Indirect field for GITS_BASER6 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER6-type

Type field for GITS_BASER6 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER7-entry-bytes

Number of bytes required per entry for GITS_BASER7 register.

Type: `int`. Default value: `0x8`.

GITS_BASER7-indirect-RAZ

Indirect field for GITS_BASER7 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER7-type

Type field for GITS_BASER7 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_PIDR

The value for the GITS_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `int`. Default value: `0x0`.

has-two-security-states

If true, has two security states.
Type: bool. Default value: 0x1.

has_mpam

Enable MPAM support on ITS and RDs.
Type: bool. Default value: 0x0.

has_VPENDBASER-dirty-flag-on-load

GICR_VPENDBASER.Dirty reflects transient loading state when valid=1.
Type: bool. Default value: 0x0.

ICFGR-PPI-mask

Mask for writes to ICFGR registers that configure PPIs.
Type: int. Default value: 0xaaaaaaaa.

ICFGR-PPI-reset

Reset value for ICFGR registers that configure PPIs.
Type: int. Default value: 0x0.

ICFGR-rsvd-bit

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGRn for n>0.
Type: bool. Default value: 0x0.

ICFGR-SGI-mask

Mask for writes to ICFGR registers that configure SGIs.
Type: int. Default value: 0x0.

ICFGR-SGI-reset

Reset value for ICFGR registers that configure SGIs.
Type: int. Default value: 0xaaaaaaaa.

ICFGR-SPI-mask

Mask for writes to ICFGR registers that configure SPIs.
Type: int. Default value: 0xaaaaaaaa.

ICFGR-SPI-reset

Reset value for ICFGR registers that configure SPIs.
Type: int. Default value: 0x0.

ignore-generate-sgi-when-no-are

Ignore GenerateSGL packets coming from the CPU interface if both ARE_S and ARE_NS are 0.
Type: bool. Default value: 0x0.

IGROUP-PPI-mask

Mask for writes to PPI bits in IGROUP registers.
Type: int. Default value: 0xffff.

IGROUP-PPI-reset

Reset value for SGI bits in IGROUP registers.
Type: int. Default value: 0x0.

IGROUP-SGI-mask

Mask for writes to SGI bits in IGROUP registers.

Type: `int`. Default value: `0xffff`.

IGROUP-SGI-reset

Reset value for SGI bits in IGROUP registers.

Type: `int`. Default value: `0x0`.

IIDR

GICD_IIDR and GICR_IIDR value.

Type: `int`. Default value: `0x0`.

IRI-ID-bits

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.

Type: `int`. Default value: `0x10`.

irouter-default-mask

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

Type: `string`. Default value: `""`.

irouter-default-reset

Default Reset Value of IROUTER[32..1019] register in the form 'a.b.c.d' or '*'.

Type: `string`. Default value: `""`.

IROUTER-IRM-RAZ-WI

GICD_IROUTERn.InterruptRoutingMode is RAZ/WI.

Type: `bool`. Default value: `0x0`.

irouter-mask-values

Mask Value of IROUTER[n] register in the form 'n=a.b.c.d'. n can be ≥ 32 and ≤ 1019 .

Type: `string`. Default value: `""`.

irouter-reset-values

Reset Value of IROUTER[n] register in the form 'n=a.b.c.d or n=*'. n can be ≥ 32 and ≤ 1019 .

Type: `string`. Default value: `""`.

ITS-BASER-force-page-alignment

Force alignment of address written to a GITS_BASER register to the page size configured.

Type: `bool`. Default value: `0x1`.

ITS-cache-invalidate-on-disable

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>.

Type: `bool`. Default value: `0x0`.

ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x0`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x0`.

ITS-cumulative-collection-tables

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`.

Type: `bool`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-entry-size

Number of bytes required to store each entry in the ITT tables.

Type: `int`. Default value: `0x8`.

ITS-hardware-collection-count

Number of hardware collections held exclusively in the ITS.

Type: `int`. Default value: `0x0`.

ITS-ID-bits

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

ITS-legacy-iidr-typer-offset

Put the `GITS_IIDR` and `GITS_TYPER` registers at their older offset of `0x8` and `0x4` respectively.

Type: `bool`. Default value: `0x0`.

ITS-MOVALL-update-collections

Whether `MOVALL` command updates the collection entires.

Type: `bool`. Default value: `0x0`.

ITS-shared-vpe-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `int`. Default value: `0x0`.

ITS-threaded-command-queue

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

ITS-TRANSLATE64R

Add an implementation specific register at `0x10008` supporting 64 bit `TRANSLATER` (`dev[63:32]`, `interrupt[31:0]`).

Type: `bool`. Default value: `0x0`.

ITS-use-physical-target-addresses

Use physical hardware addresses for targets in ITS commands -- must be true for distributed implementations.

Type: `bool`. Default value: `0x1`.

ITS-vmovp-bit

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.

Type: `bool`. Default value: `0x0`.

ITS0-base

Register base address for ITS0 (automatic if 0).

Type: `int`. Default value: `0x0`.

ITS1-base

Register base address for ITS1 (automatic if 0).

Type: `int`. Default value: `0x0`.

ITS2-base

Register base address for ITS2 (automatic if 0).

Type: `int`. Default value: `0x0`.

ITS3-base

Register base address for ITS3 (automatic if 0).

Type: `int`. Default value: `0x0`.

legacy-sgi-enable-rao

Enables for SGI associated with an ARE=0 regime are RAO/WI.

Type: `bool`. Default value: `0x0`.

local-SEIs

Generate SEI to signal internal issues.

Type: `bool`. Default value: `0x0`.

local-VSEIs

Generate VSEI to signal internal issues.

Type: `bool`. Default value: `0x0`.

lockable-SPI-count

Number of SPIs that are locked down when CFGSDISABLE signal is asserted. Only applies for GICv2.

Type: `int`. Default value: `0x0`.

LPI-cache-check-data

Enable Cached LPI data against memory checking when available for cache type.

Type: `bool`. Default value: `0x0`.

LPI-cache-type

Cache type for LPIs, 0:No caching, 1:Full caching.

Type: `int`. Default value: `0x1`.

max-pe-on-chip

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

monolithic

Indicate that the implementation is not distributed.

Type: `bool`. Default value: `0x0`.

mpam_max_partid

Maximum valid PARTID.

Type: `int`. Default value: `0xffff`.

mpam_max_pmg

Maximum valid PMG.

Type: `int`. Default value: `0xff`.

MSI_IIDR

Value returned in MSI_IIDR registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame0-base

If non-zero, sets the base address used for non-secure MSI frame 0 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame0-max-SPI

Maximum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame0-min-SPI

Minimum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame1-base

If non-zero, sets the base address used for non-secure MSI frame 1 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame1-max-SPI

Maximum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame1-min-SPI

Minimum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame2-base

If non-zero, sets the base address used for non-secure MSI frame 2 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame2-max-SPI

Maximum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame2-min-SPI

Minimum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame3-base

If non-zero, sets the base address used for non-secure MSI frame 3 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame3-max-SPI

Maximum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame3-min-SPI

Minimum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame4-base

If non-zero, sets the base address used for non-secure MSI frame 4 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame4-max-SPI

Maximum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame4-min-SPI

Minimum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame5-base

If non-zero, sets the base address used for non-secure MSI frame 5 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame5-max-SPI

Maximum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame5-min-SPI

Minimum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame6-base

If non-zero, sets the base address used for non-secure MSI frame 6 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame6-max-SPI

Maximum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame6-min-SPI

Minimum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame7-base

If non-zero, sets the base address used for non-secure MSI frame 7 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame7-max-SPI

Maximum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame7-min-SPI

Minimum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_PIDR

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.

Type: `int`. Default value: `0x0`.

MSI_S-frame0-base

If non-zero, sets the base address used for secure MSI frame 0 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame0-max-SPI

Maximum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame0-min-SPI

Minimum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame1-base

If non-zero, sets the base address used for secure MSI frame 1 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame1-max-SPI

Maximum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame1-min-SPI

Minimum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame2-base

If non-zero, sets the base address used for secure MSI frame 2 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame2-max-SPI

Maximum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame2-min-SPI

Minimum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame3-base

If non-zero, sets the base address used for secure MSI frame 3 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame3-max-SPI

Maximum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame3-min-SPI

Minimum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame4-base

If non-zero, sets the base address used for secure MSI frame 4 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame4-max-SPI

Maximum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame4-min-SPI

Minimum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame5-base

If non-zero, sets the base address used for secure MSI frame 5 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame5-max-SPI

Maximum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame5-min-SPI

Minimum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame6-base

If non-zero, sets the base address used for secure MSI frame 6 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame6-max-SPI

Maximum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame6-min-SPI

Minimum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame7-base

If non-zero, sets the base address used for secure MSI frame 7 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame7-max-SPI

Maximum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame7-min-SPI

Minimum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if `enable-multichip-operation` is set.

Type: `bool`. Default value: `0x1`.

non-ARE-core-count

Maximum number of non-ARE cores; normally used to pass the cluster-level `NUM_CORES` parameter to the top-level redistributor.

Type: `int`. Default value: `0x8`.

outer-cacheability-support

Allow configuration of outer cacheability attributes in ITS and Redistributor.

Type: `bool`. Default value: `0x0`.

output_attributes

User-defined transform to be applied to bus attributes like `MasterID`, `ExtendedID` or `UserFlags`. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`. Default value: `"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS"`.

PA_SIZE

Number of valid bits in physical address.

Type: `int`. Default value: `0x30`.

PPI-implemented-mask

Mask of PPIs that are implemented. One bit per PPI bit 0 == PPI 16 (first PPI). This will affect other masks.

Type: `int`. Default value: `0xffff`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

priority-bits

Number of implemented priority bits.

Type: `int`. Default value: `0x5`.

processor-numbers

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.

Type: `string`. Default value: `""`.

redistributor-threaded-sync

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

reg-base

Base for decoding GICv3 registers.

Type: `int`. Default value: `0x2c010000`.

reg-base-per-redistributor

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.

Type: `string`. Default value: `""`.

reg-base-per-redistributor-file

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.

Type: `string`. Default value: `""`.

report-MSI-error-via-statusr

Report MSI error via GITS_STATUSR. (0:unsupported, 1:report by GITS_STATUSR, 2:report by GITS_STATUSR and interrupt as well).

Type: `int`. Default value: `0x0`.

sgi-range-selector-support

Device has support for the Range Selector feature for SGI.

Type: `bool`. Default value: `0x0`.

single-set-support

When true, forces redistributors to recall interrupts with a clear rather than issue a second Set command.

Type: `bool`. Default value: `0x0`.

SPI-count

Number of SPIs that are implemented.

Type: `int`. Default value: `0xe0`.

SPI-message-based-support

Distributor supports message based signaling of SPI.

Type: `bool`. Default value: `0x1`.

SPI-unimplemented

A comma separated list of unimplemented SPIs ranges for sparse SPI definition(for ex: '35, 39-42, 73').

Type: `string`. Default value: `""`.

STATUSR-implemented

Determines whether the GICR_STATUSR register is implemented.

Type: `bool`. Default value: `0x1`.

supports-shareability

Device supports shareability attributes on outgoing memory bus (i.e. is modelling an ACElite port rather than an AXI4 port).

Type: `bool`. Default value: `0x1`.

trace-speculative-lpi-property-update

Trace LPI property updates performed on speculative accesses (useful for debugging LPI).
Type: `bool`. Default value: `0x0`.

virtual-lpi-support

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.
Type: `bool`. Default value: `0x0`.

virtual-priority-bits

Number of implemented virtual priority bits.
Type: `int`. Default value: `0x5`.

wakeup-on-reset

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.
Type: `bool`. Default value: `0x0`.

4.10.40 GICv3IRI

GICv3 metacomponent for redistribution of interrupts(contains Distributor, and configurable numbers 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 4-426: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GICv3IRI contains the following CADI targets:

- GICv3IRI

GICv3IRI contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)

About GICv3IRI

The GICv3IRI 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 : GICv3IRI(
    "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 GICv3IRI

Table 4-427: Ports

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	Disable some SPIs signal.
extended_ppi_in_<n>[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu <n>, where <n> is in the range 0-255.
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_<n>[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu <n>, where n is in the range 0-255.

Name	Protocol	Type	Description
pvbuss_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbuss_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Resets.
spi_in[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 GICv3IRI

A3-affinity-supported

Device supports affinity level 3 values that are non-zero.

Type: `bool`. Default value: `0x0`.

add-output-cpu-wake-request-signal-from-redistributor

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.

Type: `bool`. Default value: `0x0`.

allow-LPIEN-clear

Allow RW behaviour on `GICR_CTLR.LPIEN` instead of set once.

Type: `bool`. Default value: `0x0`.

ARE-fixed-to-one

GICv2 compatibility is not supported and `GICD_CTLR.ARE_*` is always one.

Type: `bool`. Default value: `0x0`.

chip-count

The total number of chips supported.

Type: `int`. Default value: `0x10`.

chip-id

Chip ID when multichip operation is enabled.

Type: `int`. Default value: `0x0`.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the `ISPENDR` register.

Type: `bool`. Default value: `0x0`.

common-lpi-configuration

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.

Type: `int`. Default value: `0x0`.

common-vPE-table-affinity

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.

Type: `string`. Default value: `""`.

consolidators

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.

Type: `string`. Default value: `""`.

CPU-affinities

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.

Type: `string`. Default value: `""`.

CPU-affinities-file

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.

Type: `string`. Default value: `""`.

cross-chip-AMBA-is-ACE

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.

Type: `bool`. Default value: `0x0`.

delay-ITS-accesses

Delay accesses from the ITS until `GICR_SYNC` is read.

Type: `bool`. Default value: `0x1`.

delay-redistributor-accesses

Delay memory accesses from the redistributor until `GICR_SYNC` is read.

Type: `bool`. Default value: `0x1`.

direct-lpi-support

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: `0x0`.

DPG-ARE-only

Limit application of DPG bits to interrupt groups for which `ARE=1`.

Type: `bool`. Default value: `0x0`.

DPG-bits-implemented

Enable implementation of interrupt group participation bits or DPG bits in GICR_CTLR.

Type: `bool`. Default value: `0x0`.

DS-fixed-to-zero

Enable/disable support of single security state.

Type: `bool`. Default value: `0x0`.

enable-local-cross-chip-addressing

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.

Type: `bool`. Default value: `0x0`.

enable-multichip-operation

Enables multi-chip operation between Distributors in distributed GIC IRI.

Type: `bool`. Default value: `0x0`.

enable_protocol_checking

Enable/disable protocol checking at cpu interface.

Type: `bool`. Default value: `0x0`.

enabled

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

extended-ppi-count

Number of extended PPI supported.

Type: `int`. Default value: `0x0`.

extended-spi-count

Number of extended SPI supported.

Type: `int`. Default value: `0x0`.

fixed-routed-spis

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 .

Type: `string`. Default value: `""`.

GICD-alias

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).

Type: `int`. Default value: `0x0`.

GICD-legacy-registers-as-reserved

When ARE is RAO/WI, makes superfluous registers in GICD reserved (including for the purpose of STATUSR updates).

Type: `bool`. Default value: `0x0`.

GICD_CTLR-DS-1-means-secure-only

If GICD_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: bool. Default value: 0x0.

GICD_ITARGETSR-RAZWI

If true, the GICD_ITARGETS registers are RAZ/WI.

Type: bool. Default value: 0x0.

GICD_PIDR

The value for the GICD_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: int. Default value: 0x0.

GICR-clear-enable-supported

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.

Type: bool. Default value: 0x0.

GICR-invalidate-registers-implemented

When true, the registers GICR_INVLPIR, GICR_INVALLR and GICR_SYNCR are implemented.

Type: bool. Default value: 0x0.

GICR_PIDR

The value for the GICR_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: int. Default value: 0x0.

GICR_PROPBASER-read-only

GICR_PROPBASER register is read-only.

Type: bool. Default value: 0x0.

GICR_PROPBASER-reset-value

Value of GICR_PROPBASER on reset.

Type: int. Default value: 0x0.

gicv2-only

If true, when using the GICv3 model, pretend to be a GICv2 system.

Type: bool. Default value: 0x0.

GITS_BASER0-entry-bytes

Number of bytes required per entry for GITS_BASER0 register.

Type: int. Default value: 0x8.

GITS_BASER0-indirect-RAZ

Indirect field for GITS_BASER0 register is RAZ/WI.

Type: bool. Default value: 0x0.

GITS_BASER0-type

Type field for GITS_BASER0 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: int. Default value: 0x0.

GITS_BASER1-entry-bytes

Number of bytes required per entry for GITS_BASER1 register.

Type: `int`. Default value: `0x8`.

GITS_BASER1-indirect-RAZ

Indirect field for GITS_BASER1 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER1-type

Type field for GITS_BASER1 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER2-entry-bytes

Number of bytes required per entry for GITS_BASER2 register.

Type: `int`. Default value: `0x8`.

GITS_BASER2-indirect-RAZ

Indirect field for GITS_BASER2 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER2-type

Type field for GITS_BASER2 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER3-entry-bytes

Number of bytes required per entry for GITS_BASER3 register.

Type: `int`. Default value: `0x8`.

GITS_BASER3-indirect-RAZ

Indirect field for GITS_BASER3 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER3-type

Type field for GITS_BASER3 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER4-entry-bytes

Number of bytes required per entry for GITS_BASER4 register.

Type: `int`. Default value: `0x8`.

GITS_BASER4-indirect-RAZ

Indirect field for GITS_BASER4 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER4-type

Type field for GITS_BASER4 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER5-entry-bytes

Number of bytes required per entry for GITS_BASER5 register.

Type: `int`. Default value: `0x8`.

GITS_BASER5-indirect-RAZ

Indirect field for GITS_BASER5 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER5-type

Type field for GITS_BASER5 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER6-entry-bytes

Number of bytes required per entry for GITS_BASER6 register.

Type: `int`. Default value: `0x8`.

GITS_BASER6-indirect-RAZ

Indirect field for GITS_BASER6 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER6-type

Type field for GITS_BASER6 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER7-entry-bytes

Number of bytes required per entry for GITS_BASER7 register.

Type: `int`. Default value: `0x8`.

GITS_BASER7-indirect-RAZ

Indirect field for GITS_BASER7 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER7-type

Type field for GITS_BASER7 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_PIDR

The value for the GITS_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `int`. Default value: `0x0`.

has-two-security-states

If true, has two security states.

Type: `bool`. Default value: `0x1`.

has_mpam

Enable MPAM support on ITS and RDs.

Type: `bool`. Default value: `0x0`.

has_VPENDBASER-dirty-flag-on-load

GICR_VPENDBASER.Dirty reflects transient loading state when valid=1.
Type: bool. Default value: 0x0.

ICFGR-PPI-mask

Mask for writes to ICFGR registers that configure PPIs.
Type: int. Default value: 0xaaaaaaaa.

ICFGR-PPI-reset

Reset value for ICFGR registers that configure PPIs.
Type: int. Default value: 0x0.

ICFGR-rsvd-bit

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGRn for n>0.
Type: bool. Default value: 0x0.

ICFGR-SGI-mask

Mask for writes to ICFGR registers that configure SGIs.
Type: int. Default value: 0x0.

ICFGR-SGI-reset

Reset value for ICFGR registers that configure SGIs.
Type: int. Default value: 0xaaaaaaaa.

ICFGR-SPI-mask

Mask for writes to ICFGR registers that configure SPIs.
Type: int. Default value: 0xaaaaaaaa.

ICFGR-SPI-reset

Reset value for ICFGR registers that configure SPIs.
Type: int. Default value: 0x0.

ignore-generate-sgi-when-no-are

Ignore GenerateSgi packets coming from the CPU interface if both ARE_S and ARE_NS are 0.
Type: bool. Default value: 0x0.

IGROUP-PPI-mask

Mask for writes to PPI bits in IGROUP registers.
Type: int. Default value: 0xfffff.

IGROUP-PPI-reset

Reset value for SGI bits in IGROUP registers.
Type: int. Default value: 0x0.

IGROUP-SGI-mask

Mask for writes to SGI bits in IGROUP registers.
Type: int. Default value: 0xfffff.

IGROUP-SGI-reset

Reset value for SGI bits in IGROUP registers.
Type: int. Default value: 0x0.

IIDR

GICD_IIDR and GICR_IIDR value.

Type: `int`. Default value: `0x0`.

IRI-ID-bits

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.

Type: `int`. Default value: `0x10`.

irouter-default-mask

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

Type: `string`. Default value: `""`.

irouter-default-reset

Default Reset Value of IROUTER[32..1019] register in the form 'a.b.c.d' or '*'.

Type: `string`. Default value: `""`.

IROUTER-IRM-RAZ-WI

GICD_IROUTERn.InterruptRoutingMode is RAZ/WI.

Type: `bool`. Default value: `0x0`.

irouter-mask-values

Mask Value of IROUTER[n] register in the form 'n=a.b.c.d'. n can be ≥ 32 and ≤ 1019 .

Type: `string`. Default value: `""`.

irouter-reset-values

Reset Value of IROUTER[n] register in the form 'n=a.b.c.d' or 'n=*'. n can be ≥ 32 and ≤ 1019 .

Type: `string`. Default value: `""`.

ITS-BASER-force-page-alignment

Force alignment of address written to a GITS_BASER register to the page size configured.

Type: `bool`. Default value: `0x1`.

ITS-cache-invalidate-on-disable

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>.

Type: `bool`. Default value: `0x0`.

ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x0`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x0`.

ITS-cumulative-collection-tables

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.

Type: `bool`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-entry-size

Number of bytes required to store each entry in the ITT tables.

Type: `int`. Default value: `0x8`.

ITS-hardware-collection-count

Number of hardware collections held exclusively in the ITS.

Type: `int`. Default value: `0x0`.

ITS-ID-bits

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

ITS-legacy-iidr-typer-offset

Put the GITS_IIDR and GITS_TYPER registers at their older offset of 0x8 and 0x4 respectively.

Type: `bool`. Default value: `0x0`.

ITS-MOVALL-update-collections

Whether MOVALL command updates the collection entires.

Type: `bool`. Default value: `0x0`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

Type: `int`. Default value: `0x0`.

ITS-threaded-command-queue

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

ITS-TRANSLATE64R

Add an implementation specific register at 0x10008 supporting 64 bit TRANSLATER (dev[63:32], interrupt[31:0]).

Type: `bool`. Default value: `0x0`.

ITS-use-physical-target-addresses

Use physical hardware addresses for targets in ITS commands -- must be true for distributed implementations.

Type: `bool`. Default value: `0x1`.

ITS-vmovp-bit

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.

Type: `bool`. Default value: `0x0`.

ITS0-base

Register base address for ITS0 (automatic if 0).

Type: `int`. Default value: `0x0`.

ITS1-base

Register base address for ITS1 (automatic if 0).

Type: `int`. Default value: `0x0`.

ITS2-base

Register base address for ITS2 (automatic if 0).

Type: `int`. Default value: `0x0`.

ITS3-base

Register base address for ITS3 (automatic if 0).

Type: `int`. Default value: `0x0`.

legacy-sgi-enable-rao

Enables for SGI associated with an ARE=0 regime are RAO/WI.

Type: `bool`. Default value: `0x0`.

local-SEIs

Generate SEI to signal internal issues.

Type: `bool`. Default value: `0x0`.

local-VSEIs

Generate VSEI to signal internal issues.

Type: `bool`. Default value: `0x0`.

lockable-SPI-count

Number of SPIs that are locked down when CFGSDISABLE signal is asserted. Only applies for GICv2.

Type: `int`. Default value: `0x0`.

LPI-cache-check-data

Enable Cached LPI data against memory checking when available for cache type.

Type: `bool`. Default value: `0x0`.

LPI-cache-type

Cache type for LPIs, 0:No caching, 1:Full caching.

Type: `int`. Default value: `0x1`.

max-pe-on-chip

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

monolithic

Indicate that the implementation is not distributed.

Type: `bool`. Default value: `0x0`.

mpam_max_partid

Maximum valid PARTID.

Type: `int`. Default value: `0xffff`.

mpam_max_pmg

Maximum valid PMG.

Type: `int`. Default value: `0xff`.

MSI_IIDR

Value returned in MSI_IIDR registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame0-base

If non-zero, sets the base address used for non-secure MSI frame 0 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame0-max-SPI

Maximum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame0-min-SPI

Minimum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame1-base

If non-zero, sets the base address used for non-secure MSI frame 1 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame1-max-SPI

Maximum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame1-min-SPI

Minimum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame2-base

If non-zero, sets the base address used for non-secure MSI frame 2 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame2-max-SPI

Maximum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame2-min-SPI

Minimum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame3-base

If non-zero, sets the base address used for non-secure MSI frame 3 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame3-max-SPI

Maximum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame3-min-SPI

Minimum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame4-base

If non-zero, sets the base address used for non-secure MSI frame 4 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame4-max-SPI

Maximum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame4-min-SPI

Minimum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame5-base

If non-zero, sets the base address used for non-secure MSI frame 5 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame5-max-SPI

Maximum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame5-min-SPI

Minimum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame6-base

If non-zero, sets the base address used for non-secure MSI frame 6 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame6-max-SPI

Maximum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame6-min-SPI

Minimum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame7-base

If non-zero, sets the base address used for non-secure MSI frame 7 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame7-max-SPI

Maximum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame7-min-SPI

Minimum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_PIDR

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.

Type: `int`. Default value: `0x0`.

MSI_S-frame0-base

If non-zero, sets the base address used for secure MSI frame 0 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame0-max-SPI

Maximum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame0-min-SPI

Minimum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame1-base

If non-zero, sets the base address used for secure MSI frame 1 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame1-max-SPI

Maximum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame1-min-SPI

Minimum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame2-base

If non-zero, sets the base address used for secure MSI frame 2 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame2-max-SPI

Maximum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame2-min-SPI

Minimum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame3-base

If non-zero, sets the base address used for secure MSI frame 3 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame3-max-SPI

Maximum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame3-min-SPI

Minimum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame4-base

If non-zero, sets the base address used for secure MSI frame 4 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame4-max-SPI

Maximum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame4-min-SPI

Minimum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame5-base

If non-zero, sets the base address used for secure MSI frame 5 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame5-max-SPI

Maximum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame5-min-SPI

Minimum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame6-base

If non-zero, sets the base address used for secure MSI frame 6 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame6-max-SPI

Maximum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame6-min-SPI

Minimum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame7-base

If non-zero, sets the base address used for secure MSI frame 7 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame7-max-SPI

Maximum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame7-min-SPI

Minimum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if `enable-multichip-operation` is set.

Type: `bool`. Default value: `0x1`.

non-ARE-core-count

Maximum number of non-ARE cores; normally used to pass the cluster-level `NUM_CORES` parameter to the top-level redistributor.

Type: `int`. Default value: `0x8`.

outer-cacheability-support

Allow configuration of outer cacheability attributes in ITS and Redistributor.

Type: `bool`. Default value: `0x0`.

output_attributes

User-defined transform to be applied to bus attributes like `MasterID`, `ExtendedID` or `UserFlags`. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`. Default value: `"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS"`.

PA_SIZE

Number of valid bits in physical address.

Type: `int`. Default value: `0x30`.

PPI-implemented-mask

Mask of PPIs that are implemented. One bit per PPI bit 0 == PPI 16 (first PPI). This will affect other masks.

Type: `int`. Default value: `0xffff`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

priority-bits

Number of implemented priority bits.

Type: `int`. Default value: `0x5`.

processor-numbers

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.

Type: `string`. Default value: `""`.

redistributor-threaded-sync

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

reg-base

Base for decoding GICv3 registers.

Type: `int`. Default value: `0x2c010000`.

reg-base-per-redistributor

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.

Type: `string`. Default value: `""`.

reg-base-per-redistributor-file

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.

Type: `string`. Default value: `""`.

report-MSI-error-via-statusr

Report MSI error via GITS_STATUSR. (0:unsupported, 1:report by GITS_STATUSR, 2:report by GITS_STATUSR and interrupt as well).

Type: `int`. Default value: `0x0`.

sgi-range-selector-support

Device has support for the Range Selector feature for SGI.

Type: `bool`. Default value: `0x0`.

single-set-support

When true, forces redistributors to recall interrupts with a clear rather than issue a second Set command.

Type: `bool`. Default value: `0x0`.

SPI-count

Number of SPIs that are implemented.

Type: `int`. Default value: `0xe0`.

SPI-message-based-support

Distributor supports message based signaling of SPI.

Type: `bool`. Default value: `0x1`.

SPI-unimplemented

A comma separated list of unimplemented SPIs ranges for sparse SPI definition(for ex: '35, 39-42, 73').

Type: `string`. Default value: `""`.

STATUSR-implemented

Determines whether the GICR_STATUSR register is implemented.

Type: `bool`. Default value: `0x1`.

supports-shareability

Device supports shareability attributes on outgoing memory bus (i.e. is modelling an ACElite port rather than an AXI4 port).

Type: `bool`. Default value: `0x1`.

trace-speculative-lpi-property-update

Trace LPI property updates performed on speculative accesses (useful for debugging LPI).
Type: `bool`. Default value: `0x0`.

virtual-lpi-support

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.
Type: `bool`. Default value: `0x0`.

virtual-priority-bits

Number of implemented virtual priority bits.
Type: `int`. Default value: `0x5`.

wakeup-on-reset

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.
Type: `bool`. Default value: `0x0`.

4.10.41 GICv3IRI_Filter

GICv3 metacomponent for redistribution of interrupts(contains Distributor, and configurable numbers 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 4-428: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

GICv3IRI_Filter contains the following CADI targets:

- GICv3IRI

GICv3IRI_Filter contains the following MTI components:

- [GICv3Distributor](#)
- [GICv3IRI](#)
- [GICv3Redistributor](#)
- [GICv3RedistributorInternal](#)

About GICv3IRI_Filter

The GICv3IRI_Filter has similar behavior to the GICv3IRI, except for the slave interface. Any transaction accessing a 4KB page that is not used by the GIC, as configurable through the parameters, is forwarded to the `pvtbus_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 GICv3IRI_Filter

Table 4-429: Ports

Name	Protocol	Type	Description
<code>cfgsdisable</code>	Signal	Slave	Disable some SPIs signal.
<code>extended_ppi_in_<n>[64]</code>	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu <n>, where <n> is in the range 0-255.
<code>extended_spi_in[1024]</code>	Signal	Slave	Extended Shared peripheral interrupts.
<code>msi_error_interrupt</code>	Signal	Master	When report of MSI error allowed through interrupt, loop-back this signal to relevant IRQ input signal
<code>po_reset</code>	Signal	Slave	Resets.
<code>ppi_in_<n>[16]</code>	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu <n>, where n is in the range 0-255.
<code>pvbus_filtermiss_m</code>	PVBus	Master	Passthrough for accesses to pages not used by the GIC IRI.
<code>pvbus_m</code>	PVBus	Master	Memory bus for transactions generated by the GIC.
<code>pvbus_s</code>	PVBus	Slave	Memory bus in.
<code>redistributor_m[256]</code>	GICv3Comms	Master	Input from and output to CPU interface.
<code>reset</code>	Signal	Slave	Resets.
<code>spi_in[988]</code>	Signal	Slave	Shared peripheral interrupts.
<code>wake_request[256]</code>	Signal	Master	Power management outputs.
<code>wire_to_msi_in_0[1024]</code>	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 0.
<code>wire_to_msi_in_1[1024]</code>	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 1.
<code>wire_to_msi_in_2[1024]</code>	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 2.
<code>wire_to_msi_in_3[1024]</code>	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 3.

Parameters for GICv3IRI

A3-affinity-supported

Device supports affinity level 3 values that are non-zero.

Type: `bool`. Default value: `0x0`.

add-output-cpu-wake-request-signal-from-redistributor

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.

Type: `bool`. Default value: `0x0`.

allow-LPIEN-clear

Allow RW behaviour on GICR_CTLR.LPIEN instead of set once.

Type: `bool`. Default value: `0x0`.

ARE-fixed-to-one

GICv2 compatibility is not supported and GICD_CTLR.ARE_* is always one.

Type: `bool`. Default value: `0x0`.

chip-count

The total number of chips supported.

Type: `int`. Default value: `0x10`.

chip-id

Chip ID when multichip operation is enabled.

Type: `int`. Default value: `0x0`.

clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`. Default value: `0x0`.

common-lpi-configuration

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).

Type: `int`. Default value: `0x0`.

common-vPE-table-affinity

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.

Type: `string`. Default value: `""`.

consolidators

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.

Type: `string`. Default value: `""`.

CPU-affinities

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.

Type: `string`. Default value: `""`.

CPU-affinities-file

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.

Type: `string`. Default value: `""`.

cross-chip-AMBA-is-ACE

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.

Type: `bool`. Default value: `0x0`.

delay-ITS-accesses

Delay accesses from the ITS until GICR_SYNCRR is read.

Type: `bool`. Default value: `0x1`.

delay-redistributor-accesses

Delay memory accesses from the redistributor until GICR_SYNCRR is read.

Type: `bool`. Default value: `0x1`.

direct-lpi-support

Enable support for LPI operations through GICR registers.

Type: `bool`. Default value: `0x0`.

DPG-ARE-only

Limit application of DPG bits to interrupt groups for which ARE=1.

Type: `bool`. Default value: `0x0`.

DPG-bits-implemented

Enable implementation of interrupt group participation bits or DPG bits in GICR_CTLR.

Type: `bool`. Default value: `0x0`.

DS-fixed-to-zero

Enable/disable support of single security state.

Type: `bool`. Default value: `0x0`.

enable-local-cross-chip-addressing

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.

Type: `bool`. Default value: `0x0`.

enable-multichip-operation

Enables multi-chip operation between Distributors in distributed GIC IRI.

Type: `bool`. Default value: `0x0`.

enable_protocol_checking

Enable/disable protocol checking at cpu interface.

Type: `bool`. Default value: `0x0`.

enabled

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`. Default value: `0x1`.

extended-ppi-count

Number of extended PPI supported.

Type: `int`. Default value: `0x0`.

extended-spi-count

Number of extended SPI supported.

Type: `int`. Default value: `0x0`.

fixed-routed-spis

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 .

Type: `string`. Default value: `""`.

GICD-alias

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).

Type: `int`. Default value: `0x0`.

GICD-legacy-registers-as-reserved

When ARE is RAO/WI, makes superfluous registers in GICD reserved (including for the purpose of STATUSR updates).

Type: `bool`. Default value: `0x0`.

GICD_CTLR-DS-1-means-secure-only

If GICD_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: `bool`. Default value: `0x0`.

GICD_ITARGETSR-RAZWI

If true, the GICD_ITARGETS registers are RAZ/WI.

Type: `bool`. Default value: `0x0`.

GICD_PIDR

The value for the GICD_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `int`. Default value: `0x0`.

GICR-clear-enable-supported

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.

Type: `bool`. Default value: `0x0`.

GICR-invalidate-registers-implemented

When true, the registers GICR_INVLPIR, GICR_INVALLR and GICR_SYNCRR are implemented.

Type: `bool`. Default value: `0x0`.

GICR_PIDR

The value for the GICR_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `int`. Default value: `0x0`.

GICR_PROPBASER-read-only

GICR_PROPBASER register is read-only.

Type: `bool`. Default value: `0x0`.

GICR_PROPBASER-reset-value

Value of GICR_PROPBASER on reset.

Type: `int`. Default value: `0x0`.

gicv2-only

If true, when using the GICv3 model, pretend to be a GICv2 system.

Type: `bool`. Default value: `0x0`.

GITS_BASER0-entry-bytes

Number of bytes required per entry for GITS_BASER0 register.

Type: `int`. Default value: `0x8`.

GITS_BASER0-indirect-RAZ

Indirect field for GITS_BASER0 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER0-type

Type field for GITS_BASER0 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER1-entry-bytes

Number of bytes required per entry for GITS_BASER1 register.

Type: `int`. Default value: `0x8`.

GITS_BASER1-indirect-RAZ

Indirect field for GITS_BASER1 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER1-type

Type field for GITS_BASER1 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER2-entry-bytes

Number of bytes required per entry for GITS_BASER2 register.

Type: `int`. Default value: `0x8`.

GITS_BASER2-indirect-RAZ

Indirect field for GITS_BASER2 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER2-type

Type field for GITS_BASER2 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER3-entry-bytes

Number of bytes required per entry for GITS_BASER3 register.

Type: `int`. Default value: `0x8`.

GITS_BASER3-indirect-RAZ

Indirect field for GITS_BASER3 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER3-type

Type field for GITS_BASER3 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER4-entry-bytes

Number of bytes required per entry for GITS_BASER4 register.

Type: `int`. Default value: `0x8`.

GITS_BASER4-indirect-RAZ

Indirect field for GITS_BASER4 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER4-type

Type field for GITS_BASER4 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER5-entry-bytes

Number of bytes required per entry for GITS_BASER5 register.

Type: `int`. Default value: `0x8`.

GITS_BASER5-indirect-RAZ

Indirect field for GITS_BASER5 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER5-type

Type field for GITS_BASER5 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER6-entry-bytes

Number of bytes required per entry for GITS_BASER6 register.

Type: `int`. Default value: `0x8`.

GITS_BASER6-indirect-RAZ

Indirect field for GITS_BASER6 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER6-type

Type field for GITS_BASER6 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_BASER7-entry-bytes

Number of bytes required per entry for GITS_BASER7 register.

Type: `int`. Default value: `0x8`.

GITS_BASER7-indirect-RAZ

Indirect field for GITS_BASER7 register is RAZ/WI.

Type: `bool`. Default value: `0x0`.

GITS_BASER7-type

Type field for GITS_BASER7 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `int`. Default value: `0x0`.

GITS_PIDR

The value for the GITS_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `int`. Default value: `0x0`.

has-two-security-states

If true, has two security states.

Type: `bool`. Default value: `0x1`.

has_mpam

Enable MPAM support on ITS and RDs.

Type: `bool`. Default value: `0x0`.

has_VPENDBASER-dirty-flag-on-load

GICR_VPENDBASER.Dirty reflects transient loading state when valid=1.

Type: `bool`. Default value: `0x0`.

ICFGR-PPI-mask

Mask for writes to ICFGR registers that configure PPIs.

Type: `int`. Default value: `0xaaaaaaaa`.

ICFGR-PPI-reset

Reset value for ICFGR registers that configure PPIs.

Type: `int`. Default value: `0x0`.

ICFGR-rsvd-bit

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGRn for n>0.

Type: `bool`. Default value: `0x0`.

ICFGR-SGI-mask

Mask for writes to ICFGR registers that configure SGIs.

Type: `int`. Default value: `0x0`.

ICFGR-SGI-reset

Reset value for ICFGR registers that configure SGIs.

Type: `int`. Default value: `0xaaaaaaaa`.

ICFGR-SPI-mask

Mask for writes to ICFGR registers that configure SPIs.

Type: `int`. Default value: `0xaaaaaaaa`.

ICFGR-SPI-reset

Reset value for ICFGR registers that configure SPIs.

Type: `int`. Default value: `0x0`.

ignore-generate-sgi-when-no-are

Ignore GenerateSGL packets coming from the CPU interface if both ARE_S and ARE_NS are 0.

Type: `bool`. Default value: `0x0`.

IGROUP-PPI-mask

Mask for writes to PPI bits in IGROUP registers.

Type: `int`. Default value: `0xffff`.

IGROUP-PPI-reset

Reset value for SGI bits in IGROUP registers.

Type: `int`. Default value: `0x0`.

IGROUP-SGI-mask

Mask for writes to SGI bits in IGROUP registers.

Type: `int`. Default value: `0xffff`.

IGROUP-SGI-reset

Reset value for SGI bits in IGROUP registers.

Type: `int`. Default value: `0x0`.

IIDR

GICD_IIDR and GICR_IIDR value.

Type: `int`. Default value: `0x0`.

IRI-ID-bits

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.

Type: `int`. Default value: `0x10`.

irouter-default-mask

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

Type: `string`. Default value: `""`.

irouter-default-reset

Default Reset Value of IROUTER[32..1019] register in the form 'a.b.c.d' or '*'.

Type: `string`. Default value: `""`.

IROUTER-IRM-RAZ-WI

GICD_IROUTERn.InterruptRoutingMode is RAZ/WI.

Type: `bool`. Default value: `0x0`.

irouter-mask-values

Mask Value of IROUTER[n] register in the form 'n=a.b.c.d'. n can be ≥ 32 and ≤ 1019 .

Type: `string`. Default value: `""`.

irouter-reset-values

Reset Value of IROUTER[n] register in the form 'n=a.b.c.d or n=*'. n can be ≥ 32 and ≤ 1019 .

Type: `string`. Default value: `""`.

ITS-BASER-force-page-alignment

Force alignment of address written to a GITS_BASER register to the page size configured.

Type: `bool`. Default value: `0x1`.

ITS-cache-invalidate-on-disable

Sets the RO bit GITS_TYPER.INV. When true, after the following sequence: 1) GITS_CTLR.Enabled written 1-->0, 2) GITS_CTLR.Quiescent observed as 1, 3) GITS_BASER<n>.Valid written 1-->0, there is no cached information from the ITS memory structure pointed to by GITS_BASER<n>.

Type: `bool`. Default value: `0x0`.

ITS-collection-ID-bits

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS_TYPER.CIL=0).

Type: `int`. Default value: `0x0`.

ITS-count

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `int`. Default value: `0x0`.

ITS-cumulative-collection-tables

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.

Type: `bool`. Default value: `0x1`.

ITS-device-bits

Number of bits supported for ITS device IDs.

Type: `int`. Default value: `0x10`.

ITS-entry-size

Number of bytes required to store each entry in the ITT tables.

Type: `int`. Default value: `0x8`.

ITS-hardware-collection-count

Number of hardware collections held exclusively in the ITS.

Type: `int`. Default value: `0x0`.

ITS-ID-bits

Number of interrupt bits supported by ITS.

Type: `int`. Default value: `0x10`.

ITS-legacy-iidr-typer-offset

Put the GITS_IIDR and GITS_TYPER registers at their older offset of 0x8 and 0x4 respectively.

Type: `bool`. Default value: `0x0`.

ITS-MOVALL-update-collections

Whether MOVALL command updates the collection entires.

Type: `bool`. Default value: `0x0`.

ITS-shared-vPE-table

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

Type: `int`. Default value: `0x0`.

ITS-threaded-command-queue

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: `bool`. Default value: `0x1`.

ITS-TRANSLATE64R

Add an implementation specific register at 0x10008 supporting 64 bit TRANSLATER (dev[63:32], interrupt[31:0]).

Type: `bool`. Default value: `0x0`.

ITS-use-physical-target-addresses

Use physical hardware addresses for targets in ITS commands -- must be true for distributed implementations.

Type: `bool`. Default value: `0x1`.

ITS-vmovp-bit

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.

Type: `bool`. Default value: `0x0`.

ITS0-base

Register base address for ITS0 (automatic if 0).

Type: `int`. Default value: `0x0`.

ITS1-base

Register base address for ITS1 (automatic if 0).

Type: `int`. Default value: `0x0`.

ITS2-base

Register base address for ITS2 (automatic if 0).

Type: `int`. Default value: `0x0`.

ITS3-base

Register base address for ITS3 (automatic if 0).

Type: `int`. Default value: `0x0`.

legacy-sgi-enable-rao

Enables for SGI associated with an ARE=0 regime are RAO/WI.

Type: `bool`. Default value: `0x0`.

local-SEIs

Generate SEI to signal internal issues.

Type: `bool`. Default value: `0x0`.

local-VSEIs

Generate VSEI to signal internal issues.

Type: `bool`. Default value: `0x0`.

lockable-SPI-count

Number of SPIs that are locked down when CFGSDISABLE signal is asserted. Only applies for GICv2.

Type: `int`. Default value: `0x0`.

LPI-cache-check-data

Enable Cached LPI data against memory checking when available for cache type.

Type: `bool`. Default value: `0x0`.

LPI-cache-type

Cache type for LPIs, 0:No caching, 1:Full caching.

Type: `int`. Default value: `0x1`.

max-pe-on-chip

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`. Default value: `0x4`.

monolithic

Indicate that the implementation is not distributed.

Type: `bool`. Default value: `0x0`.

mpam_max_partid

Maximum valid PARTID.

Type: `int`. Default value: `0xffff`.

mpam_max_pmg

Maximum valid PMG.

Type: `int`. Default value: `0xff`.

MSI_IIDR

Value returned in MSI_IIDR registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame0-base

If non-zero, sets the base address used for non-secure MSI frame 0 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame0-max-SPI

Maximum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame0-min-SPI

Minimum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame1-base

If non-zero, sets the base address used for non-secure MSI frame 1 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame1-max-SPI

Maximum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame1-min-SPI

Minimum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame2-base

If non-zero, sets the base address used for non-secure MSI frame 2 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame2-max-SPI

Maximum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame2-min-SPI

Minimum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame3-base

If non-zero, sets the base address used for non-secure MSI frame 3 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame3-max-SPI

Maximum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame3-min-SPI

Minimum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame4-base

If non-zero, sets the base address used for non-secure MSI frame 4 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame4-max-SPI

Maximum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame4-min-SPI

Minimum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame5-base

If non-zero, sets the base address used for non-secure MSI frame 5 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame5-max-SPI

Maximum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame5-min-SPI

Minimum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame6-base

If non-zero, sets the base address used for non-secure MSI frame 6 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame6-max-SPI

Maximum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame6-min-SPI

Minimum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame7-base

If non-zero, sets the base address used for non-secure MSI frame 7 registers.

Type: `int`. Default value: `0x0`.

MSI_NS-frame7-max-SPI

Maximum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_NS-frame7-min-SPI

Minimum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_PIDR

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.

Type: `int`. Default value: `0x0`.

MSI_S-frame0-base

If non-zero, sets the base address used for secure MSI frame 0 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame0-max-SPI

Maximum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame0-min-SPI

Minimum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame1-base

If non-zero, sets the base address used for secure MSI frame 1 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame1-max-SPI

Maximum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame1-min-SPI

Minimum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame2-base

If non-zero, sets the base address used for secure MSI frame 2 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame2-max-SPI

Maximum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame2-min-SPI

Minimum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame3-base

If non-zero, sets the base address used for secure MSI frame 3 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame3-max-SPI

Maximum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame3-min-SPI

Minimum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame4-base

If non-zero, sets the base address used for secure MSI frame 4 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame4-max-SPI

Maximum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame4-min-SPI

Minimum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame5-base

If non-zero, sets the base address used for secure MSI frame 5 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame5-max-SPI

Maximum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame5-min-SPI

Minimum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame6-base

If non-zero, sets the base address used for secure MSI frame 6 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame6-max-SPI

Maximum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame6-min-SPI

Minimum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame7-base

If non-zero, sets the base address used for secure MSI frame 7 registers.

Type: `int`. Default value: `0x0`.

MSI_S-frame7-max-SPI

Maximum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

MSI_S-frame7-min-SPI

Minimum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `int`. Default value: `0x0`.

multichip-threaded-dgi

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if `enable-multichip-operation` is set.

Type: `bool`. Default value: `0x1`.

non-ARE-core-count

Maximum number of non-ARE cores; normally used to pass the cluster-level `NUM_CORES` parameter to the top-level redistributor.

Type: `int`. Default value: `0x8`.

outer-cacheability-support

Allow configuration of outer cachability attributes in ITS and Redistributor.

Type: `bool`. Default value: `0x0`.

output_attributes

User-defined transform to be applied to bus attributes like `MasterID`, `ExtendedID` or `UserFlags`. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`. Default value: `"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS"`.

PA_SIZE

Number of valid bits in physical address.

Type: `int`. Default value: `0x30`.

PPI-implemented-mask

Mask of PPIs that are implemented. One bit per PPI bit 0 == PPI 16 (first PPI). This will affect other masks.

Type: `int`. Default value: `0xfffff`.

print-memory-map

Print memory map to stdout.

Type: `bool`. Default value: `0x0`.

priority-bits

Number of implemented priority bits.

Type: `int`. Default value: `0x5`.

processor-numbers

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.

Type: string. Default value: "".

redistributor-threaded-sync

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: bool. Default value: 0x1.

reg-base

Base for decoding GICv3 registers.

Type: int. Default value: 0x2c010000.

reg-base-per-redistributor

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.

Type: string. Default value: "".

reg-base-per-redistributor-file

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.

Type: string. Default value: "".

report-MSI-error-via-statusr

Report MSI error via GITS_STATUSR. (0:unsupported, 1:report by GITS_STATUSR, 2:report by GITS_STATUSR and interrupt as well).

Type: int. Default value: 0x0.

sgi-range-selector-support

Device has support for the Range Selector feature for SGI.

Type: bool. Default value: 0x0.

single-set-support

When true, forces redistributors to recall interrupts with a clear rather than issue a second Set command.

Type: bool. Default value: 0x0.

SPI-count

Number of SPIs that are implemented.

Type: int. Default value: 0xe0.

SPI-message-based-support

Distributor supports message based signaling of SPI.

Type: bool. Default value: 0x1.

SPI-unimplemented

A comma separated list of unimplemented SPIs ranges for sparse SPI definition(for ex: '35, 39-42, 73').

Type: `string`. Default value: `""`.

STATUSR-implemented

Determines whether the GICR_STATUSR register is implemented.

Type: `bool`. Default value: `0x1`.

supports-shareability

Device supports shareability attributes on outgoing memory bus (i.e. is modelling an ACElite port rather than an AXI4 port).

Type: `bool`. Default value: `0x1`.

trace-speculative-lpi-property-update

Trace LPI property updates performed on speculative accesses (useful for debugging LPI).

Type: `bool`. Default value: `0x0`.

virtual-lpi-support

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.

Type: `bool`. Default value: `0x0`.

virtual-priority-bits

Number of implemented virtual priority bits.

Type: `int`. Default value: `0x5`.

wakeup-on-reset

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.

Type: `bool`. Default value: `0x0`.

4.10.42 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 4-430: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

ICS307 contains the following CADI targets:

- ClockDivider
- ICS307

ICS307 contains the following MTI components:

- [ClockDivider](#)

About ICS307

You can use this component to convert the rate of one ClockSignal to another ClockSignal by using configurable multiplier, divider, and scale values. The divider ratio can be set by startup parameters or at runtime by a configuration port. Changes to the input ClockSignal rate and divider ratio are reflected immediately by the output ClockSignal ports.

Three values determine the divisor ratio:

- `vdw`.
- `rdw`.
- `od`.

To calculate the divisor ratio, use:

$$\text{Divisor} = ((\text{rdw}+2) * \text{scale}) / (2 * (\text{vdw}+8))$$

where `scale` is derived from this table indexed by `od`:

Table 4-431: 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 4-432: Ports

Name	Protocol	Type	Description
<code>clk_in</code>	ClockSignal	Slave	Master clock rate.
<code>clk_out_clk1</code>	ClockSignal	Master	Modified clock rate.
<code>clk_out_ref</code>	ClockSignal	Master	Pass through of master clock rate for divider chaining.
<code>configuration</code>	ICS307Configuration	Slave	Configuration port for setting divider ratio dynamically.

Parameters for ICS307

od

OD.
Type: `int`. Default value: `0x3`.

rdr

RDR.
Type: `int`. Default value: `0x6`.

vdw

VDW.
Type: `int`. Default value: `0x4`.

4.10.43 IDAU

IDAU is device which provides Security attribute relating to the address pass 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 4-433: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

About IDAU

An Implementation Defined Attribution Unit (*IDAU*) is a device that provides a security attribute relating to an address passed to it. For each memory access (data and instructions), the CPU checks the IDAU and sets the security of its transactions based on it.

The IDAU model uses the `pv::IDAUSignal` struct to return the security attributes for the address that is 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.

The fields of the `pv::IDAUSignal` struct map to the hardware signals as follows:

Table 4-434: Mappings between `pv::IDAUSignal` fields and hardware signals

Hardware signal	<code>pv::IDAUSignal</code> field	Description
IDAUNS	<code>bool ns</code>	Non-secure region response.
IDAUNSC	<code>bool nsc</code>	Non-secure-callable region response.
IDAUID	<code>uint8_t region</code>	Region number.
IDAUIDV	<code>bool valid</code>	Region number valid.
IDAUNCHK	<code>bool exempt</code>	Region exempt from attribution check.

Masters can read or write to the `pvbuss_s` port as follows:

Read

Reads return the IDAU region's `pv::IdauRegion` struct (32 bytes), which contains 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

The port only supports 32-byte write operations to pass in a `pv::IdauRegion` struct for updating an internal IDAU region.

DMI

The 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.



To disable the IDAU, set the `NUM_IDAU_REGION` parameter to zero.

Ports for IDAU

Table 4-435: Ports

Name	Protocol	Type	Description
<code>invalidate_region</code>	<code>Value_64</code>	Master	This port is used as a call back to inform masters that the IDAU has updated its region information.
<code>pvbuss_s</code>	PVBus	Slave	-

4.10.44 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 4-436: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

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, <https://www.jedec.org>) 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 2GB.

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_SELDL_CARD.
- MMC_SEND_CSD.
- MMC_SEND_CID.
- MMC_STOP_TRANSMISSION.
- MMC_SEND_STATUS.

- `MMC_GO_INACTIVE_STATE.`
- `MMC_READ_SINGLE_BLOCK.`
- `MMC_READ_MULTIPLE_BLOCK.`
- `MMC_SET_BLOCK_COUNT.`
- `MMC_WRITE_BLOCK.`
- `MMC_WRITE_MULTIPLE_BLOCK.`
- `MMC_SEND_EXT_CSD.` This command is supported in SDHC mode only.

The block length is 512 bytes. SimGen reports attempts to change it as errors.

The component supports these erase commands (Class 5), but they have no effect on the disk backing storage:

- `MMC_ERASE_GROUP_START.`
- `MMC_ERASE_GROUP_END.`
- `MMC_ERASE.`

The component does not support these commands:

- `MMC_BUSTEST_R.`
- `MMC_BUSTEST_W.`

The component does not support stream read and write commands (Classes 1 and 3):

- `MMC_READ_DAT_UNTIL_STOP.`
- `MMC_WRITE_DAT_UNTIL_STOP.`
- `MMC_PROGRAM_CID.`
- `MMC_PROGRAM_CSD.`

The component does not support block oriented write protection commands (Class 6):

- `MMC_SET_WRITE_PROT.`
- `MMC_CLR_WRITE_PROT.`
- `MMC_SEND_WRITE_PROT.`

The component does not support lock card commands (Class 7) or application-specific commands (Class 8):

- `MMC_LOCK_UNLOCK.`
- `MMC_APP_CMD.`
- `MMC_GEN_CMD.`

The component does not support I/O mode commands (Class 9):

- `MMC_FAST_IO.`

- `MMC_GO_IRQ_STATE`.

The component does not support reserved commands. Using a reserved command sets the `MMC_ST_ER_B_ILLEGAL_COMMAND` bit in the status register of the card. Read this with the `MMC_SEND_STATUS` command.

Use the `p_diagnostics` parameter to select the level of diagnostic output, to help to debug device driver and controller-to-card protocol issues. It supports the following levels:

Level 0

None.

Level 1

Warnings about attempting to change read-only settings.

Level 2

Trace of command calls.

Level 3

Information about every step in the `MMC_Protocol` interaction.

Level 4

Hex dump of every block sent or received.

The registers are not memory mapped. Instead, you access them using relevant MMC commands. The MMC component model makes the registers available through a CADI interface. Modification of these registers through CADI is not recommended, but not prohibited. For example, modifying the card ID (CID) registers can be useful when experimenting with drivers, but direct modification of the `STATUS_REG` register is likely to put the card model into an indeterminate state.

For a full definition of MMC registers, see the MMCA System Summary documentation. Device-specific register information can also be obtained from MMC vendors.

Table 4-437: 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

Name	CADI register number	Description
BLOCK_COUNT_REG	0x010	Block count

Ports for MMC

Table 4-438: 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

Card type ('SD' or 'SDHC').

Type: `string`. Default value: "SDHC".

diagnostics

Diagnostics level.

Type: `int`. Default value: 0x0.

force_sector_addressing

Use sector addressing even on small cards.

Type: `bool`. Default value: 0x0.

p_fast_access

Don't simulate MMC block access delays.

Type: `bool`. Default value: 0x1.

p_manid

Card ID Manufacturer ID.

Type: `int`. Default value: 0x2.

p_max_block_count

Default maximum block count reg. Default 0x80.

Type: `int`. Default value: 0x80.

p_mmc_file

MMCard filename.

Type: `string`. Default value: "mmc.dat".

p_OEMid

Card ID OEM ID.

Type: `int`. Default value: 0x0.

p_prodName

Card ID Product Name (6 chars).

Type: `string`. Default value: "ARMmmc".

p_prodRev

Card ID Product Revision.

Type: `int`. Default value: `0x1`.

p_sernum

Card Serial Number.
Type: `int`. Default value: `0xca4d0001`.

support_unpadded_images

Support images that are not a multiple of 512k by padding them to the needed size (SDHC cards only).
Type: `bool`. Default value: `0x0`.

4.10.45 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 4-439: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

MMU_400 contains the following CADI targets:

- MMU_400
- MMU_400_BASE

MMU_400 contains the following MTI components:

- [MMU_400_BASE](#)
- [PVBusMapper](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

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 4-440: 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_cdtw_in	Signal	Slave	Enables coherent page table walks.
cfgflt_irpt_ns	Signal	Master	Non-secure configuration access fault interrupt. Corresponds to SMMU architectural signal SMMU_NSgCgflrpt.
cfgflt_irpt_s	Signal	Master	Secure configuration access fault interrupt. Corresponds to SMMU architectural signal SMMU_gCgflrpt.
comb_irpt_ns	Signal	Master	Non-secure combined interrupt.
comb_irpt_s	Signal	Master	Secure combined interrupt.
cxt_irpt_ns	Signal	Master	Non-secure context bank fault.
glblflt_irpt_ns	Signal	Master	Global Non-secure fault interrupt. Corresponds to SMMU architectural signal SMMU_NSglrpt.
glblflt_irpt_s	Signal	Master	Global Secure fault interrupt. Corresponds to SMMU architectural signal SMMU_glrpt.
priv_internals	MMU_400_Internals	Slave	For internal use only, please do not use.
pvbus_m	PVBus	Master	Downstream port of the MMU, where translated transactions emerge.

Name	Protocol	Type	Description
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

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

Type: string. Default value: "".

cfg_cttw

Perform coherent page table walks.

Type: bool. Default value: 0x1.

dump_unpredictability_in_user_flags

Override the user flags to encode unpredictable information (validation only).

Type: bool. Default value: 0x0.

label0_read_ssd

Label0: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label0_read_stream_id

Label0: Read Stream ID.

Type: int. Default value: 0x0.

label0_write_ssd

Label0: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label0_write_stream_id

Label0: Write Stream ID.

Type: int. Default value: 0x0.

label10_read_ssd

Label10: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label10_read_stream_id

Label10: Read Stream ID.

Type: int. Default value: 0x0.

label10_write_ssd

Label10: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label10_write_stream_id

Label10: Write Stream ID.

Type: int. Default value: 0x0.

label11_read_ssd

Label11: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label11_read_stream_id

Label11: Read Stream ID.
Type: int. Default value: 0x0.

label11_write_ssd

Label11: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label11_write_stream_id

Label11: Write Stream ID.
Type: int. Default value: 0x0.

label12_read_ssd

Label12: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label12_read_stream_id

Label12: Read Stream ID.
Type: int. Default value: 0x0.

label12_write_ssd

Label12: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label12_write_stream_id

Label12: Write Stream ID.
Type: int. Default value: 0x0.

label13_read_ssd

Label13: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label13_read_stream_id

Label13: Read Stream ID.
Type: int. Default value: 0x0.

label13_write_ssd

Label13: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label13_write_stream_id

Label13: Write Stream ID.
Type: int. Default value: 0x0.

label14_read_ssd

Label14: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label14_read_stream_id

Label14: Read Stream ID.

Type: int. Default value: 0x0.

label14_write_ssd

Label14: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label14_write_stream_id

Label14: Write Stream ID.

Type: int. Default value: 0x0.

label15_read_ssd

Label15: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label15_read_stream_id

Label15: Read Stream ID.

Type: int. Default value: 0x0.

label15_write_ssd

Label15: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label15_write_stream_id

Label15: Write Stream ID.

Type: int. Default value: 0x0.

label16_read_ssd

Label16: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label16_read_stream_id

Label16: Read Stream ID.

Type: int. Default value: 0x0.

label16_write_ssd

Label16: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label16_write_stream_id

Label16: Write Stream ID.

Type: int. Default value: 0x0.

label17_read_ssd

Label17: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label17_read_stream_id

Label17: Read Stream ID.

Type: int. Default value: 0x0.

label17_write_ssd

Label17: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label17_write_stream_id

Label17: Write Stream ID.
Type: int. Default value: 0x0.

label18_read_ssd

Label18: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label18_read_stream_id

Label18: Read Stream ID.
Type: int. Default value: 0x0.

label18_write_ssd

Label18: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label18_write_stream_id

Label18: Write Stream ID.
Type: int. Default value: 0x0.

label19_read_ssd

Label19: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label19_read_stream_id

Label19: Read Stream ID.
Type: int. Default value: 0x0.

label19_write_ssd

Label19: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label19_write_stream_id

Label19: Write Stream ID.
Type: int. Default value: 0x0.

label1_read_ssd

Label1: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label1_read_stream_id

Label1: Read Stream ID.
Type: int. Default value: 0x0.

label1_write_ssd

Label1: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label1_write_stream_id

Label1: Write Stream ID.

Type: int. Default value: 0x0.

label20_read_ssd

Label20: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label20_read_stream_id

Label20: Read Stream ID.

Type: int. Default value: 0x0.

label20_write_ssd

Label20: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label20_write_stream_id

Label20: Write Stream ID.

Type: int. Default value: 0x0.

label21_read_ssd

Label21: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label21_read_stream_id

Label21: Read Stream ID.

Type: int. Default value: 0x0.

label21_write_ssd

Label21: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label21_write_stream_id

Label21: Write Stream ID.

Type: int. Default value: 0x0.

label22_read_ssd

Label22: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label22_read_stream_id

Label22: Read Stream ID.

Type: int. Default value: 0x0.

label22_write_ssd

Label22: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label22_write_stream_id

Label22: Write Stream ID.

Type: int. Default value: 0x0.

label23_read_ssd

Label23: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label23_read_stream_id

Label23: Read Stream ID.
Type: int. Default value: 0x0.

label23_write_ssd

Label23: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label23_write_stream_id

Label23: Write Stream ID.
Type: int. Default value: 0x0.

label24_read_ssd

Label24: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label24_read_stream_id

Label24: Read Stream ID.
Type: int. Default value: 0x0.

label24_write_ssd

Label24: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label24_write_stream_id

Label24: Write Stream ID.
Type: int. Default value: 0x0.

label25_read_ssd

Label25: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label25_read_stream_id

Label25: Read Stream ID.
Type: int. Default value: 0x0.

label25_write_ssd

Label25: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label25_write_stream_id

Label25: Write Stream ID.
Type: int. Default value: 0x0.

label26_read_ssd

Label26: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label26_read_stream_id

Label26: Read Stream ID.

Type: int. Default value: 0x0.

label26_write_ssd

Label26: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label26_write_stream_id

Label26: Write Stream ID.

Type: int. Default value: 0x0.

label27_read_ssd

Label27: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label27_read_stream_id

Label27: Read Stream ID.

Type: int. Default value: 0x0.

label27_write_ssd

Label27: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label27_write_stream_id

Label27: Write Stream ID.

Type: int. Default value: 0x0.

label28_read_ssd

Label28: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label28_read_stream_id

Label28: Read Stream ID.

Type: int. Default value: 0x0.

label28_write_ssd

Label28: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label28_write_stream_id

Label28: Write Stream ID.

Type: int. Default value: 0x0.

label29_read_ssd

Label29: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label29_read_stream_id

Label29: Read Stream ID.

Type: int. Default value: 0x0.

label29_write_ssd

Label29: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label29_write_stream_id

Label29: Write Stream ID.
Type: int. Default value: 0x0.

label2_read_ssd

Label2: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label2_read_stream_id

Label2: Read Stream ID.
Type: int. Default value: 0x0.

label2_write_ssd

Label2: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label2_write_stream_id

Label2: Write Stream ID.
Type: int. Default value: 0x0.

label30_read_ssd

Label30: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label30_read_stream_id

Label30: Read Stream ID.
Type: int. Default value: 0x0.

label30_write_ssd

Label30: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label30_write_stream_id

Label30: Write Stream ID.
Type: int. Default value: 0x0.

label31_read_ssd

Label31: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label31_read_stream_id

Label31: Read Stream ID.
Type: int. Default value: 0x0.

label31_write_ssd

Label31: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label31_write_stream_id

Label31: Write Stream ID.

Type: `int`. Default value: `0x0`.**label3_read_ssd**

Label3: Read SDD or SSD_Index.

Type: `int`. Default value: `0x0`.**label3_read_stream_id**

Label3: Read Stream ID.

Type: `int`. Default value: `0x0`.**label3_write_ssd**

Label3: Write SDD or SSD_Index.

Type: `int`. Default value: `0x0`.**label3_write_stream_id**

Label3: Write Stream ID.

Type: `int`. Default value: `0x0`.**label4_read_ssd**

Label4: Read SDD or SSD_Index.

Type: `int`. Default value: `0x0`.**label4_read_stream_id**

Label4: Read Stream ID.

Type: `int`. Default value: `0x0`.**label4_write_ssd**

Label4: Write SDD or SSD_Index.

Type: `int`. Default value: `0x0`.**label4_write_stream_id**

Label4: Write Stream ID.

Type: `int`. Default value: `0x0`.**label5_read_ssd**

Label5: Read SDD or SSD_Index.

Type: `int`. Default value: `0x0`.**label5_read_stream_id**

Label5: Read Stream ID.

Type: `int`. Default value: `0x0`.**label5_write_ssd**

Label5: Write SDD or SSD_Index.

Type: `int`. Default value: `0x0`.**label5_write_stream_id**

Label5: Write Stream ID.

Type: `int`. Default value: `0x0`.

label6_read_ssd

Label6: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label6_read_stream_id

Label6: Read Stream ID.
Type: int. Default value: 0x0.

label6_write_ssd

Label6: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label6_write_stream_id

Label6: Write Stream ID.
Type: int. Default value: 0x0.

label7_read_ssd

Label7: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label7_read_stream_id

Label7: Read Stream ID.
Type: int. Default value: 0x0.

label7_write_ssd

Label7: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label7_write_stream_id

Label7: Write Stream ID.
Type: int. Default value: 0x0.

label8_read_ssd

Label8: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label8_read_stream_id

Label8: Read Stream ID.
Type: int. Default value: 0x0.

label8_write_ssd

Label8: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label8_write_stream_id

Label8: Write Stream ID.
Type: int. Default value: 0x0.

label9_read_ssd

Label9: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label9_read_stream_id

Label9: Read Stream ID.

Type: `int`. Default value: `0x0`.**label9_write_ssd**

Label9: Write SSD or SSD_Index.

Type: `int`. Default value: `0x0`.**label9_write_stream_id**

Label9: Write Stream ID.

Type: `int`. Default value: `0x0`.**number_of_contexts**

Number of context banks.

Type: `int`. Default value: `0x8`.**number_of_smrs**

Number of stream match registers.

Type: `int`. Default value: `0x20`.**percent_tlbstatus_commits**

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

Type: `int`. Default value: `0xa`.**prefetch_only_requests**

Handle prefetch-only requests by:- 0 -- deny them 1 -- use debug table walks and TLB entries 2 -- treat them as normal transactions (dangerous).

Type: `int`. Default value: `0x0`.**programmable_non_secure_by_default_ssd_indices**

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

Type: `string`. Default value: `""`.**programmable_secure_by_default_ssd_indices**

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

Type: `string`. Default value: `""`.**ptw_has_separate_port**Page Table Walks use `pvbus_ptw_m`.Type: `bool`. Default value: `0x1`.**pvbus_m_is_ace_lite**Is `pvbus_m` (the downstream port that translated transaction exit) ACE-Lite.Type: `bool`. Default value: `0x1`.**pvbus_ptw_m_is_ace_lite**Is `pvbus_ptw_m` (the downstream port that is used for walks if `ptw_has_separate_port` is true) ACE-Lite.Type: `bool`. Default value: `0x1`.**stream_id_width**

StreamID bit width.

Type: `int`. Default value: `0x6`.

tlb_depth

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

Type: `int`. Default value: `0x40`.

use_label_mapping

Use label mapping.

Type: `bool`. Default value: `0x1`.

use_ssd_determination_table

Use SSD Determination Table.

Type: `bool`. Default value: `0x1`.

Parameters for MMU_400_BASE

mmu.always_secure_ssd_indices

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

Type: `string`. Default value: `""`.

mmu.cfg_cttw

Perform coherent page table walks.

Type: `bool`. Default value: `0x1`.

mmu.dump_unpredictability_in_user_flags

Override the user flags to encode unpredictable information (validation only).

Type: `bool`. Default value: `0x0`.

mmu.number_of_contexts

Number of context banks.

Type: `int`. Default value: `0x8`.

mmu.number_of_smrs

Number of stream match registers.

Type: `int`. Default value: `0x10`.

mmu.percent_tlbstatus_commits

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

Type: `int`. Default value: `0xa`.

mmu.prefetch_only_requests

Handle prefetch-only requests by: 0 -- deny them 1 -- use debug table walks and TLB entries 2 -- treat them as normal transactions (dangerous).

Type: `int`. Default value: `0x0`.

mmu.programmable_non_secure_by_default_ssd_indices

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

Type: `string`. Default value: `""`.

mmu.programmable_secure_by_default_ssd_indices

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

Type: `string`. Default value: `""`.

mmu.ptw_has_separate_port

Page Table Walks use `pvbus_ptw_m`.

Type: `bool`. Default value: `0x1`.

`mmu.pvbus_m_is_ace_lite`

Is `pvbus_m` (the downstream port that translated transaction exit) ACE-Lite.

Type: `bool`. Default value: `0x1`.

`mmu.pvbus_ptw_m_is_ace_lite`

Is `pvbus_ptw_m` (the downstream port that is used for walks if `ptw_has_separate_port` is true) ACE-Lite.

Type: `bool`. Default value: `0x1`.

`mmu.seed`

Seed for SMMU.

Type: `int`. Default value: `0x12345678`.

`mmu.stream_id_width`

StreamID bit width.

Type: `int`. Default value: `0x6`.

`mmu.tlb_depth`

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

Type: `int`. Default value: `0x40`.

`mmu.use_ssd_determination_table`

Use SSD Determination Table.

Type: `bool`. Default value: `0x1`.

4.10.46 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 4-441: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

MMU_400_BASE contains the following CADI targets:

- MMU_400_BASE

MMU_400_BASE contains the following MTI components:

- [MMU_400_BASE](#)
- [PVBusMapper](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

Ports for MMU_400_BASE

Table 4-442: 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.
cfg_flt_irpt_ns	Signal	Master	Non-secure Configuration Access Fault Interrupt In the SMMU Architecture this is called SMMU_NSgCflrpt.
cfg_flt_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" (cfg_flt_irpt_ns glbl_flt_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).
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_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.

Name	Protocol	Type	Description
pvbust_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.
pvbust_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

mmu.always_secure_ssd_indices

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

Type: string. Default value: "".

mmu.cfg_cttw

Perform coherent page table walks.

Type: bool. Default value: 0x1.

mmu.dump_unpredictability_in_user_flags

Override the user flags to encode unpredictable information (validation only).

Type: bool. Default value: 0x0.

mmu.number_of_contexts

Number of context banks.

Type: int. Default value: 0x8.

mmu.number_of_smrs

Number of stream match registers.

Type: int. Default value: 0x10.

mmu.percent_tlbstatus_commits

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

Type: int. Default value: 0xa.

mmu.prefetch_only_requests

Handle prefetch-only requests by:- 0 -- deny them 1 -- use debug table walks and TLB entries 2 -- treat them as normal transactions (dangerous).

Type: int. Default value: 0x0.

mmu.programmable_non_secure_by_default_ssd_indices

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

Type: string. Default value: "".

mmu.programmable_secure_by_default_ssd_indices

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

Type: `string`. Default value: `""`.

mmu.ptw_has_separate_port

Page Table Walks use `pvbus_ptw_m`.

Type: `bool`. Default value: `0x1`.

mmu.pvbus_m_is_ace_lite

Is `pvbus_m` (the downstream port that translated transaction exit) ACE-Lite.

Type: `bool`. Default value: `0x1`.

mmu.pvbus_ptw_m_is_ace_lite

Is `pvbus_ptw_m` (the downstream port that is used for walks if `ptw_has_separate_port` is true) ACE-Lite.

Type: `bool`. Default value: `0x1`.

mmu.seed

Seed for SMMU.

Type: `int`. Default value: `0x12345678`.

mmu.stream_id_width

StreamID bit width.

Type: `int`. Default value: `0x6`.

mmu.tlb_depth

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

Type: `int`. Default value: `0x40`.

mmu.use_ssd_determination_table

Use SSD Determination Table.

Type: `bool`. Default value: `0x1`.

4.10.47 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 4-443: IP revisions support

Revision	Quality level
r2p4	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

MMU_500 contains the following CADI targets:

- MMU_500
- MMU_500_BASE

MMU_500 contains the following MTI components:

- [MMU_500_BASE](#)
- [PVBusMapper](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

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 4-444: 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

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

Type: string. Default value: "".

cfg_cttw

Perform coherent page table walks.

Type: bool. Default value: 0x1.

dump_unpredictability_in_user_flags

Override the user flags to encode unpredictable information (validation only).

Type: bool. Default value: 0x0.

label0_read_ssd

Label0: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label0_read_stream_id

Label0: Read Stream ID.

Type: int. Default value: 0x0.

label0_write_ssd

Label0: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label0_write_stream_id

Label0: Write Stream ID.

Type: int. Default value: 0x0.

label10_read_ssd

Label10: Read SDD or SSD_Index.

Type: `int`. Default value: `0x0`.

label10_read_stream_id

Label10: Read Stream ID.

Type: `int`. Default value: `0x0`.

label10_write_ssd

Label10: Write SDD or SSD_Index.

Type: `int`. Default value: `0x0`.

label10_write_stream_id

Label10: Write Stream ID.

Type: `int`. Default value: `0x0`.

label11_read_ssd

Label11: Read SDD or SSD_Index.

Type: `int`. Default value: `0x0`.

label11_read_stream_id

Label11: Read Stream ID.

Type: `int`. Default value: `0x0`.

label11_write_ssd

Label11: Write SDD or SSD_Index.

Type: `int`. Default value: `0x0`.

label11_write_stream_id

Label11: Write Stream ID.

Type: `int`. Default value: `0x0`.

label12_read_ssd

Label12: Read SDD or SSD_Index.

Type: `int`. Default value: `0x0`.

label12_read_stream_id

Label12: Read Stream ID.

Type: `int`. Default value: `0x0`.

label12_write_ssd

Label12: Write SDD or SSD_Index.

Type: `int`. Default value: `0x0`.

label12_write_stream_id

Label12: Write Stream ID.

Type: `int`. Default value: `0x0`.

label13_read_ssd

Label13: Read SDD or SSD_Index.

Type: `int`. Default value: `0x0`.

label13_read_stream_id

Label13: Read Stream ID.

Type: `int`. Default value: `0x0`.

label13_write_ssd

Label13: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label13_write_stream_id

Label13: Write Stream ID.
Type: int. Default value: 0x0.

label14_read_ssd

Label14: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label14_read_stream_id

Label14: Read Stream ID.
Type: int. Default value: 0x0.

label14_write_ssd

Label14: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label14_write_stream_id

Label14: Write Stream ID.
Type: int. Default value: 0x0.

label15_read_ssd

Label15: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label15_read_stream_id

Label15: Read Stream ID.
Type: int. Default value: 0x0.

label15_write_ssd

Label15: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label15_write_stream_id

Label15: Write Stream ID.
Type: int. Default value: 0x0.

label16_read_ssd

Label16: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label16_read_stream_id

Label16: Read Stream ID.
Type: int. Default value: 0x0.

label16_write_ssd

Label16: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label16_write_stream_id

Label16: Write Stream ID.

Type: int. Default value: 0x0.

label17_read_ssd

Label17: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label17_read_stream_id

Label17: Read Stream ID.

Type: int. Default value: 0x0.

label17_write_ssd

Label17: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label17_write_stream_id

Label17: Write Stream ID.

Type: int. Default value: 0x0.

label18_read_ssd

Label18: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label18_read_stream_id

Label18: Read Stream ID.

Type: int. Default value: 0x0.

label18_write_ssd

Label18: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label18_write_stream_id

Label18: Write Stream ID.

Type: int. Default value: 0x0.

label19_read_ssd

Label19: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label19_read_stream_id

Label19: Read Stream ID.

Type: int. Default value: 0x0.

label19_write_ssd

Label19: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label19_write_stream_id

Label19: Write Stream ID.

Type: int. Default value: 0x0.

label1_read_ssd

Label1: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label1_read_stream_id

Label1: Read Stream ID.

Type: int. Default value: 0x0.

label1_write_ssd

Label1: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label1_write_stream_id

Label1: Write Stream ID.

Type: int. Default value: 0x0.

label20_read_ssd

Label20: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label20_read_stream_id

Label20: Read Stream ID.

Type: int. Default value: 0x0.

label20_write_ssd

Label20: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label20_write_stream_id

Label20: Write Stream ID.

Type: int. Default value: 0x0.

label21_read_ssd

Label21: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label21_read_stream_id

Label21: Read Stream ID.

Type: int. Default value: 0x0.

label21_write_ssd

Label21: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label21_write_stream_id

Label21: Write Stream ID.

Type: int. Default value: 0x0.

label22_read_ssd

Label22: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label22_read_stream_id

Label22: Read Stream ID.

Type: int. Default value: 0x0.

label22_write_ssd

Label22: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label22_write_stream_id

Label22: Write Stream ID.

Type: int. Default value: 0x0.

label23_read_ssd

Label23: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label23_read_stream_id

Label23: Read Stream ID.

Type: int. Default value: 0x0.

label23_write_ssd

Label23: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label23_write_stream_id

Label23: Write Stream ID.

Type: int. Default value: 0x0.

label24_read_ssd

Label24: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label24_read_stream_id

Label24: Read Stream ID.

Type: int. Default value: 0x0.

label24_write_ssd

Label24: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label24_write_stream_id

Label24: Write Stream ID.

Type: int. Default value: 0x0.

label25_read_ssd

Label25: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label25_read_stream_id

Label25: Read Stream ID.

Type: int. Default value: 0x0.

label25_write_ssd

Label25: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label25_write_stream_id

Label25: Write Stream ID.
Type: int. Default value: 0x0.

label26_read_ssd

Label26: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label26_read_stream_id

Label26: Read Stream ID.
Type: int. Default value: 0x0.

label26_write_ssd

Label26: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label26_write_stream_id

Label26: Write Stream ID.
Type: int. Default value: 0x0.

label27_read_ssd

Label27: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label27_read_stream_id

Label27: Read Stream ID.
Type: int. Default value: 0x0.

label27_write_ssd

Label27: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label27_write_stream_id

Label27: Write Stream ID.
Type: int. Default value: 0x0.

label28_read_ssd

Label28: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label28_read_stream_id

Label28: Read Stream ID.
Type: int. Default value: 0x0.

label28_write_ssd

Label28: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label28_write_stream_id

Label28: Write Stream ID.

Type: `int`. Default value: `0x0`.**label29_read_ssd**

Label29: Read SDD or SSD_Index.

Type: `int`. Default value: `0x0`.**label29_read_stream_id**

Label29: Read Stream ID.

Type: `int`. Default value: `0x0`.**label29_write_ssd**

Label29: Write SDD or SSD_Index.

Type: `int`. Default value: `0x0`.**label29_write_stream_id**

Label29: Write Stream ID.

Type: `int`. Default value: `0x0`.**label12_read_ssd**

Label2: Read SDD or SSD_Index.

Type: `int`. Default value: `0x0`.**label12_read_stream_id**

Label2: Read Stream ID.

Type: `int`. Default value: `0x0`.**label12_write_ssd**

Label2: Write SDD or SSD_Index.

Type: `int`. Default value: `0x0`.**label12_write_stream_id**

Label2: Write Stream ID.

Type: `int`. Default value: `0x0`.**label130_read_ssd**

Label30: Read SDD or SSD_Index.

Type: `int`. Default value: `0x0`.**label130_read_stream_id**

Label30: Read Stream ID.

Type: `int`. Default value: `0x0`.**label130_write_ssd**

Label30: Write SDD or SSD_Index.

Type: `int`. Default value: `0x0`.**label130_write_stream_id**

Label30: Write Stream ID.

Type: `int`. Default value: `0x0`.

label31_read_ssd

Label31: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label31_read_stream_id

Label31: Read Stream ID.
Type: int. Default value: 0x0.

label31_write_ssd

Label31: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label31_write_stream_id

Label31: Write Stream ID.
Type: int. Default value: 0x0.

label3_read_ssd

Label3: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label3_read_stream_id

Label3: Read Stream ID.
Type: int. Default value: 0x0.

label3_write_ssd

Label3: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label3_write_stream_id

Label3: Write Stream ID.
Type: int. Default value: 0x0.

label4_read_ssd

Label4: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label4_read_stream_id

Label4: Read Stream ID.
Type: int. Default value: 0x0.

label4_write_ssd

Label4: Write SDD or SSD_Index.
Type: int. Default value: 0x0.

label4_write_stream_id

Label4: Write Stream ID.
Type: int. Default value: 0x0.

label5_read_ssd

Label5: Read SDD or SSD_Index.
Type: int. Default value: 0x0.

label5_read_stream_id

Label5: Read Stream ID.

Type: int. Default value: 0x0.

label5_write_ssd

Label5: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label5_write_stream_id

Label5: Write Stream ID.

Type: int. Default value: 0x0.

label6_read_ssd

Label6: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label6_read_stream_id

Label6: Read Stream ID.

Type: int. Default value: 0x0.

label6_write_ssd

Label6: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label6_write_stream_id

Label6: Write Stream ID.

Type: int. Default value: 0x0.

label7_read_ssd

Label7: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label7_read_stream_id

Label7: Read Stream ID.

Type: int. Default value: 0x0.

label7_write_ssd

Label7: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label7_write_stream_id

Label7: Write Stream ID.

Type: int. Default value: 0x0.

label8_read_ssd

Label8: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label8_read_stream_id

Label8: Read Stream ID.

Type: int. Default value: 0x0.

label8_write_ssd

Label8: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label8_write_stream_id

Label8: Write Stream ID.

Type: int. Default value: 0x0.

label9_read_ssd

Label9: Read SDD or SSD_Index.

Type: int. Default value: 0x0.

label9_read_stream_id

Label9: Read Stream ID.

Type: int. Default value: 0x0.

label9_write_ssd

Label9: Write SDD or SSD_Index.

Type: int. Default value: 0x0.

label9_write_stream_id

Label9: Write Stream ID.

Type: int. Default value: 0x0.

number_of_contexts

Number of context banks.

Type: int. Default value: 0x8.

number_of_smrs

Number of stream match registers.

Type: int. Default value: 0x20.

percent_tlbstatus_commits

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

Type: int. Default value: 0xa.

prefetch_only_requests

Handle prefetch-only requests by: - 0 -- deny them 1 -- use debug table walks and TLB entries 2 -- treat them as normal transactions (dangerous).

Type: int. Default value: 0x0.

programmable_non_secure_by_default_ssd_indices

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

Type: string. Default value: "".

programmable_secure_by_default_ssd_indices

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

Type: string. Default value: "".

ptw_has_separate_port

Page Table Walks use pvbus_ptw_m.

Type: bool. Default value: 0x1.

supports_nested_translations

Supports nested translations (stage 1 + stage 2).

Type: `bool`. Default value: `0x1`.

tlb_depth

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

Type: `int`. Default value: `0x800`.

use_label_mapping

Use label mapping.

Type: `bool`. Default value: `0x1`.

use_ssd_determination_table

Use SSD Determination Table.

Type: `bool`. Default value: `0x1`.

version

Version of the RTL that the model represents. Valid values are LACr1 and EAC.

Type: `string`. Default value: `"EAC"`.

Parameters for MMU_500_BASE**mmu.always_secure_ssd_indices**

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

Type: `string`. Default value: `""`.

mmu.cfg_ettw

Perform coherent page table walks.

Type: `bool`. Default value: `0x1`.

mmu.dump_unpredictability_in_user_flags

Override the user flags to encode unpredictable information (validation only).

Type: `bool`. Default value: `0x0`.

mmu.number_of_contexts

Number of context banks.

Type: `int`. Default value: `0x8`.

mmu.number_of_smrs

Number of stream match registers.

Type: `int`. Default value: `0x10`.

mmu.percent_tlbstatus_commits

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

Type: `int`. Default value: `0xa`.

mmu.prefetch_only_requests

Handle prefetch-only requests by: - 0 -- deny them 1 -- use debug table walks and TLB entries 2 -- treat them as normal transactions (dangerous).

Type: `int`. Default value: `0x0`.

mmu.PRIVATE_PARAMETER_personality

The personality to use (affects ID codes and various imp def features).

Type: `string`. Default value: `""`.

`mmu.PRIVATE_PARAMETER_seed`

Seed for randomised SMMU implementation defined behaviour.

Type: `int`. Default value: `0x12345678`.

`mmu.PRIVATE_PARAMETER_validation_mode`

Internal validation mode.

Type: `int`. Default value: `0x0`.

`mmu.programmable_non_secure_by_default_ssd_indices`

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

Type: `string`. Default value: `""`.

`mmu.programmable_secure_by_default_ssd_indices`

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

Type: `string`. Default value: `""`.

`mmu.ptw_has_separate_port`

Page Table Walks use `pvbus_ptw_m` (or uses `pvbus_m[0]`).

Type: `bool`. Default value: `0x1`.

`mmu.supports_nested_translations`

Supports nested translations (stage 1 + stage 2).

Type: `bool`. Default value: `0x1`.

`mmu.tlb_depth`

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

Type: `int`. Default value: `0x800`.

`mmu.use_ssd_determination_table`

Use SSD Determination Table.

Type: `bool`. Default value: `0x1`.

`mmu.version`

Version of the RTL that the model represents. Valid values are LACr1 and EAC.

Type: `string`. Default value: `"EAC"`.

4.10.48 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 4-445: IP revisions support

Revision	Quality level
r2p4	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

MMU_500_BASE contains the following CADI targets:

- [MMU_500_BASE](#)

MMU_500_BASE contains the following MTI components:

- [MMU_500_BASE](#)
- [PVBusMapper](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

Ports for MMU_500_BASE

Table 4-446: Ports

Name	Protocol	Type	Description
cfg_cttw_in	Signal	Slave	The SoC supports coherent page walks, this is meant to be sampled at reset. However, in practice the model has to prevent the race condition between cfg_cttw being asserted at the same 'cycle' as negedge reset. Thus we actually only sample the signal on the first transaction to the SMMU or the first transition on this signal after reset. Thus in the model, we require that cfg_cttw be held for at least this period of time.
comb_irpt_ns	Signal	Master	"Non-secure combined interrupt"
comb_irpt_s	Signal	Master	"Secure combined interrupt"
cxt_irpt[128]	Signal	Master	Non-secure context bank fault.
glblflt_irpt_ns	Signal	Master	Global non-secure Fault Interrupt In the SMMU Architecture this is called SMMU_NSGlrpt.
glblflt_irpt_s	Signal	Master	Global secure Fault Interrupt In the SMMU Architecture this is called SMMU_glrpt.
identify	MMU_500_BASE_IDENTIFY	Master	This port is a special model port that is used to take a transaction and map it to an SSD/SSD_Index and StreamID.
priv_internals	MMU_500_Internals	Slave	For internal use only, please do not use.
pvbus_control_s	PVBus	Slave	The register port of the device is AXI.
pvbus_m[32]	PVBus	Master	This downstream port is where the translated accesses from pvbus_s emerge. See notes for pvbus_s[] as well. If the Page Table Walk (PTW) does not have a separate port then PTW accesses will emerge at port 0 with the same attributes as described in pvbus_ptw_m.
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.

Name	Protocol	Type	Description
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

mmu.always_secure_ssd_indices

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

Type: string. Default value: "".

mmu.cfg_cttw

Perform coherent page table walks.

Type: bool. Default value: 0x1.

mmu.dump_unpredictability_in_user_flags

Override the user flags to encode unpredictable information (validation only).

Type: bool. Default value: 0x0.

mmu.number_of_contexts

Number of context banks.

Type: int. Default value: 0x8.

mmu.number_of_smrs

Number of stream match registers.

Type: int. Default value: 0x10.

mmu.percent_tlbstatus_commits

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

Type: int. Default value: 0xa.

mmu.prefetch_only_requests

Handle prefetch-only requests by: - 0 -- deny them 1 -- use debug table walks and TLB entries 2 -- treat them as normal transactions (dangerous).

Type: int. Default value: 0x0.

mmu.PRIVATE_PARAMETER_personality

The personality to use (affects ID codes and various imp def features).

Type: string. Default value: "".

mmu.PRIVATE_PARAMETER_seed

Seed for randomised SMMU implementation defined behaviour.

Type: int. Default value: 0x12345678.

mmu.PRIVATE_PARAMETER_validation_mode

Internal validation mode.

Type: `int`. Default value: `0x0`.

`mmu.programmable_non_secure_by_default_ssd_indices`

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

Type: `string`. Default value: `""`.

`mmu.programmable_secure_by_default_ssd_indices`

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

Type: `string`. Default value: `""`.

`mmu.ptw_has_separate_port`

Page Table Walks use `pvbus_ptw_m` (or uses `pvbus_m[0]`).

Type: `bool`. Default value: `0x1`.

`mmu.supports_nested_translations`

Supports nested translations (stage 1 + stage 2).

Type: `bool`. Default value: `0x1`.

`mmu.tlb_depth`

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

Type: `int`. Default value: `0x800`.

`mmu.use_ssd_determination_table`

Use SSD Determination Table.

Type: `bool`. Default value: `0x1`.

`mmu.version`

Version of the RTL that the model represents. Valid values are LACr1 and EAC.

Type: `string`. Default value: `"EAC"`.

4.10.49 MMU_600

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-447: IP revisions support

Revision	Quality level
r0p0	Full support
r0p1	Full support
r0p2	Full support
r1p0	Full support
r2p0	Full support
r2p1	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

MMU_600 contains the following CADI targets:

- MMU_600

MMU_600 contains the following MTI components:

- [MMU_600](#)
- [PVBusMapper](#)
- [PVBusSlave](#)

Changes in 11.24.4

Parameters added:

- `TCUCFG_XLATE_SLOTS`

About MMU_600

MMU_600 and MMU_700 are SMMUv3-compliant devices and are used for I/O virtualization of devices.

The hardware is a distributed SMMU. It consists of the following:

- A single Translation Control Unit (TCU), which has a port for the programming interface of the SMMU, receives DVM messages, and does all the page walking and queue manipulation, for example.
- One or more Translation Bus Units (TBUs), which translate transactions from upstream client devices into downstream transactions.
- Zero or more connections to PCIe Root Complexes (PCIe-RCs). A maximum of 62 TBUs and PCIe-RCs can be attached.
- An interconnect that connects the TBUs and 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, the TBUs used are 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 following 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, the `tbs_pvbus_s[i]` and `tbm_pvbus_m[i]` port pair represent a TBU *i*, or `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.

To reduce system construction complexity, the `tbs_pvbus_s[i]` and `tbm_pvbus_m[i]` pair also acts as a TBU so that the PCIe-RC does not need to 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 ATC Invalidates must be routed to the correct PCIe subsystem to invalidate the cache of ATS Responses in the subsystem. Therefore 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, 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.

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 set to one.

Debug and real accesses to the registers must be 32 bits or 64 bits.

MSIs are issued on the `qtw_pvbus_m` port using attributes that are determined by the parameter `msi_attribute_transform`, while Event queue writes are always issued with `ExtendedID=0`, `UserFlags=0`, `MasterID=0xFFFFffff`. In the hardware, there is no way of distinguishing Event queue writes from MSI writes. However, this provides a mechanism for the model system to distinguish them.

The hardware only has a single cacheability attribute for input transactions, but `PVBus` transports have both inner and outer cacheability.

For non-PCIe-mode TBUs, whose 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, and it is Write-back, 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).

Therefore, 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 value. If it is `INC-oNC-osh`, then it must be presented as such.

For PCIe-mode TBUs, that is, whose 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 is interpreted as `INC-oNC-osh`. ! NoSnoop is 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`, it is forced to `INC-oNC-osh`. For example, if a NoSnoop transaction uses a page table and is transformed to `iWB-oWB-nsh` then it is forced to `INC-oNC-osh`. However, if the page tables transformed it to a device type, then as all device types are stronger than `INC-oNC-osh`, it exits the SMMU as the device type.
- In the model, transactions are classified by their incoming memory attributes as to whether they are NoSnoop or not and then are normalized appropriately:
 - `iWB-oWB-any-shareability` are interpreted as ! NoSnoop, therefore are normalized to `iWB-oWB-ish`.
 - Anything else is considered NoSnoop, and therefore is normalized to `INC-oNC-osh`.
- Translated accesses also have the same interpretation to determine NoSnoop and how they are normalized. Therefore they could enter the system with different attributes to if they entered the SMMU as Untranslated Accesses and were translated by the normal translation process, before exiting downstream of the SMMU and entering the system
- 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, it 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), it is normalized to `iWB-oWB` with the incoming shareability.
- If the input is anything else, it is normalized to `INC-oNC-osh`.

The model accepts full architectural attributes of two levels of cacheability and so has to decide how to interpret this in terms of the hardware. For transactions that are not cache maintenance operations, the model replicates the outer attribute into the inner attribute and then performs the normalization that the hardware does.

The hardware normalizes the architectural output attributes and outputs a single level of cacheability and a user flag (OC) specifying if the architectural attributes were cacheable in the outer cacheable domain. If the transaction is classified as a PCIe transaction, the NoSnoop

transform previously described is applied. That is, if the original transaction was NoSnoop then any weaker memory type is strengthened to iNC-oNC-osh, and the following transform is applied:

```
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.



Note

The model deals with groups of transactions with the same attributes and a similar range of addresses. The mapping that is used is stored by the bus infrastructure and is used for subsequent sufficiently similar transactions without needing the intervention of the SMMU model, and therefore are not traced.

Model limitations

- The Performance Monitoring Unit (PMU) has limited functionality. It is intended for demonstration purposes only and does not implement all architecturally required events. It does not implement the `pmusnapshot_ack` or `pmusnapshot_req` interface.
- The model does not implement:
 - RAS.
 - Power control.
 - `sup_oas`, which controls the OAS of the SMMU. This is expected to be constant for a system.
 - The SYSCO interface.
 - The Low-Power Interface.
 - Any of the IMP DEF MPAM registers.
- Cache maintenance operations cannot be inserted into the TBU ports of the SMMU.
- `PvBus` has no representation of the cache stash operations, so they are not supported.
- `TCU_CFG.XLATE_SLOTS` is fixed at 512.
- `TCU_STATUS.GNT_XLATE_SLOTS` always reads as 512.

Ports for MMU_600

Table 4-448: Ports

Name	Protocol	Type	Description
<code>clk_in</code>	ClockSignal	Slave	Clock signal (in RTL <code>aclk</code>) This is a clock time-base used by the TCU to spread some of its processing over time, if enabled by the <code>wait_*</code> parameters. The clock must always be connected.
<code>cmd_sync_irpt_ns</code>	Signal	Master	Pulsed interrupt output signal for non-secure CMD_SYNC having a completion signal of <code>SIG_IRQ</code> .

Name	Protocol	Type	Description
cmd_sync_irpt_s	Signal	Master	Pulsed interrupt output signal for secure CMD_SYNC having a completion signal of SIG_IRQ.
event_q_irpt_ns	Signal	Master	Pulsed interrupt output signal for the non-secure event queue becoming non-empty.
event_q_irpt_s	Signal	Master	Pulsed interrupt output signal for the secure event queue becoming non-empty.
evento	Signal	Master	Event signal
global_irpt_ns	Signal	Master	Pulsed interrupt output signal for non-secure SMMU_GERROR(N) signalling an error.
global_irpt_s	Signal	Master	Pulsed interrupt output signal for secure SMMU_S_GERROR(N) signalling an error.
identify	SMMUv3AEMIdentifyProtocol	Master	Map the transaction to the tuple (StreamID, SubStreamID, SubStreamIDValid, SSD) The StreamID that is produced by the implementation of this protocol is not the final StreamID. The final StreamID is produced by using the list_of_ns_sid_high_at_bitpos0/list_of_s_sid_high_at_bitpos0 parameter to map the StreamID based on the upstream port index. Also see the parameter howto_identify which can replace the functionality of this port under certain circumstances.
pri_q_irpt_ns	Signal	Master	Pulsed interrupt output signal for the PRI queue becoming non-empty. 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. 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 when separate_tw_msi_qs_port=true (in RTL QTW)
sec_override	Signal	Slave	Allow certain registers to be accessible to non-secure accesses from reset, as described in the TCU_SCR register.
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.
sup_cohacc	Signal	Slave	System supports COHACC and will be reflected in the IDR registers.
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.
tbm_pvbus_m[62]	PVBus	Master	The TBU manager ports that carry transactions that have been translated from the correspondingly numbered tbs_pvbus_s[] port.
tbs_pvbus_s[62]	PVBus	Slave	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered pvbus_m[] port.
tbu_pmu_irpt[62]	Signal	Master	TBU Performance Monitoring Unit interrupt, one per TBU.
tbu_pmusnapshot_ack[62]	Signal	Master	PMU snapshot interface for the TBU, ack a snapshot.

Name	Protocol	Type	Description
tbu_pmusnapshot_req[62]	Signal	Slave	PMU snapshot interface for the TBU, request a snapshot.
tbu_ras_irpt[62]	Signal	Master	The RAS interrupt pin for errors detected in the TBUs.
tbu_reset_in[62]	Signal	Slave	Reset signals The TBUs can have independent reset signals. Each signal tbu_reset_in[n] corresponds to the TBU using tbs_pvbus_s[n]/tbs_pvbus_s[n] pair. If the SMMU receives a transaction whilst the TBU is expected to be in reset then it will complain using the ArchMsg.Warning.warning trace source. Those tbu_reset_in that correspond to a PCIe-RC connection can be connected to monitor the PCIe-RC's reset signal. If it receives an ATS request when in reset then it will complain in a similar way. You must connect these pins if you wish the TCU_NODE_STATUS for the nodes to be accurate (including any connected to the PCIe-RC).
tcu_pmu_irpt	Signal	Master	TCU Performance Monitoring Unit interrupt
tcu_pmusnapshot_ack	Signal	Master	PMU snapshot interface for the TCU, ack a snapshot.
tcu_pmusnapshot_req	Signal	Slave	PMU snapshot interface This is a four-phase handshake to have the corresponding PMCG perform a capture of the current counter values. PMU snapshot interface for the TCU, request a snapshot.
tcu_ras_irpt	Signal	Master	The RAS interrupt pin for errors detected in the TCU.
tcu_reset_in	Signal	Slave	The reset signal to the TCU interface.

Parameters for MMU_600

all_error_messages_through_trace

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. .
Type: bool. Default value: 0x0.

behaviour_of_sampled_at_reset_signals

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. .
Type: int. Default value: 0x0.

cmdq_max_number_of_commands_to_buffer

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 will accentuate this effect. .
Type: int. Default value: 0xa.

enable_device_id_checks

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 then if it overflows 32 bits then the model will warn. If the DeviceID is not used then it is assumed that the external agent that forms the DeviceID will warn if it overflows. * If the SMMU supports MSIs, then the model will check that the GIC will be 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. .

Type: bool. Default value: 0x1.

howto_identify

If 'use-identify' then will use 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. The StreamID is secure if SEC_SID is true. The SubstreamID (20 b) is valid if SSV is true. NoStreamID transactions are identified by SIDV == 0 and the SSD is the PAS of the transaction and SEC_SID is not used. nSEC_SID, nSSV, nSIDV are available with negative logic. Negative and positive logic symbols for the same attribute is a error. Different attributes are independent and can use negative or positive logic. .

Type: string. Default value: "use-identify".

list_of_ns_sid_high_at_bitpos0

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. .

Type: string. Default value: "".

list_of_pcie_mode

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 attributes handling for these TBUs are 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" .

Type: string. Default value: "".

list_of_pcie_rc

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, then the PCIe-RC uses this port for ATS/PRI, and then the actual transactions go through separate TBUs. In the model, then 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. .

Type: string. Default value: "".

list_of_s_sid_high_at_bitpos0

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. .

Type: string. Default value: "".

msi_attribute_transform

Transform downstream attributes of MSI transactions. * ""/"none" -- no transform * How to alter output attributes of SMMU-generated MSIs. Example: "UserFlags[15:0]=smmu_msi_device_id[31:16], MasterID[15:0]=smmu_msi_device_id[15:0], ExtendedID=0" RHS/LHS Symbols: * ExtendedID/MasterID/UserFlags RHS Symbols: * the parameter smmu_msi_device_id * the symbol 'interrupt_kind' * 0/1 -- EVENTQ s/ns * 2 -- PRIQ * 3/4 -- CMD_SYNC s/ns * 5/6 -- GERROR s/ns * 7/8 -- PMCG s/ns * 9/10/11 -- RAS FHI/ERI/CRI * HWATTR_KIND_0: PBHA information * numeric literals. ExtendedID/MasterID/UserFlags start with values {0, 0xFFFFFFFF, 0} respectively. Any bits with no transform are unchanged. This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices. NOTE: * see also 'output_attribute_transform' and enable_device_id_checks. .

Type: string. Default value: "ExtendedID[31:0]=smmu_msi_device_id, MasterID=0xFFFFFFFF".

number_of_ports

The number of port pairs that the SMMU has. .

Type: int. Default value: 0x1.

output_attribute_transform

Transform the downstream attributes of a translated transaction. * "" or "none" -- the input and output attributes are identical. * How to alter the output attributes, e.g. "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.

Type: string. Default value: "ExtendedID[31:0]=DeviceID".

output_id_routed_transform

The SMMU generates the following ID-routed transaction on the pvbus_id_routed_m bus:

- * ATC Invalidate
- * PRI Response This parameter expresses 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 .

Type: string. Default value: "Address[27:12]=StreamID[15:0], PAS=SSD".

prefetch_only_requests

The simulator supports 'prefetch-only' DMI requests, which can occur for anytime 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(). .

Type: int. Default value: 0x0.

sec_override

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. .

Type: bool. Default value: 0x0.

seed

Used to seed the pseudo-random number generator that the SMMU model uses. .

Type: int. Default value: 0x12345678.

size_of_cd_cache

The number of entries in the cache holding CD structures. 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. .

Type: int. Default value: 0x0.

size_of_llcd_cache

The number of entries in the cache holding L1CD descriptors. 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. .

Type: int. Default value: 0x0.

size_of_llste_cache

The number of entries in the cache holding L1STE descriptors. 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. .

Type: int. Default value: 0x0.

size_of_ste_cache

The number of entries in the cache holding STE structures. 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. .

Type: `int`. Default value: `0x0`.

size_of_tlb

The number of entries in the 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. .

Type: `int`. Default value: `0x0`.

smmu_msi_device_id

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID. See parameter `msi_attribute_transform` and `enable_device_id_checks`. .

Type: `int`. Default value: `0x0`.

sup_cohacc

The value to put in SMMU_IDR0.COHAAC .

Type: `bool`. Default value: `0x1`.

sup_oas

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 .

Type: `int`. Default value: `0x5`.

TCUCFG_XLATE_SLOTS

Maximum number of outstanding stalled transactions that the SMMU supports Note that `TCUCFG_XLATE_SLOTS` must be \geq `TCUCFG_PTW_SLOTS` which currently fixed to 512 Accepted Values: 4, 8, 16, 32, 64, 128, 256, 512 .

Type: `int`. Default value: `0x200`.

tlb_when_do_f_tlb_conflict_on_overlap

If a TLB entry is created by a walk and it overlaps an existing entry. Then 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. .

Type: `int`. Default value: `0x0`.

translated_device_id_base

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`. .

Type: `int`. Default value: `0x0`.

tw_qs_attribute_transform

Transform downstream attributes of table walk and queue transactions. *

""/"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 * numeric literals.

ExtendedID/MasterID/UserFlags start with values {0, 0xFFFFFFFF, 0} respectively. Any bits with no transform are unchanged. NOTE: * see also 'output_attribute_transform' and 'msi_attribute_transform'.

Type: string. Default value: "".

version

The version of this product.

Type: string. Default value: "r0p0".

wait_cmdq_ticks

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))).

Type: int. Default value: 0x0.

wait_eventq_ticks

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))).

Type: int. Default value: 0x0.

wait_imp_def_work_ticks

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))).

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.

Type: int. Default value: 0x0.

wait_misc_async_actions_ticks

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))).

Type: int. Default value: 0x0.

wait_msi_ticks

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))).

Type: int. Default value: 0x0.

wait_pri_req_ticks

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))).

Type: int. Default value: 0x0.

wait_pri_resp_ticks

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, that immediately makes a PRI Request, that 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))). .
Type: `int`. Default value: 0x1.

4.10.50 MMU_700

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-449: IP revisions support

Revision	Quality level
r0p0	Full support
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

MMU_700 contains the following CADI targets:

- MMU_700

MMU_700 contains the following MTI components:

- [MMU_700](#)
- [PVBusMapper](#)
- [PVBusSlave](#)

Changes in 11.24.4

Parameters added:

- `TCUCFG_XLATE_SLOTS`

About MMU_700

MMU_600 and MMU_700 are SMMUv3-compliant devices and are used for I/O virtualization of devices.

The hardware is a distributed SMMU. It consists of the following:

- A single Translation Control Unit (TCU), which has a port for the programming interface of the SMMU, receives DVM messages, and does all the page walking and queue manipulation, for example.
- One or more Translation Bus Units (TBUs), which translate transactions from upstream client devices into downstream transactions.

- Zero or more connections to PCIe Root Complexes (PCIe-RCs). A maximum of 62 TBUs and PCIe-RCs can be attached.
- An interconnect that connects the TBUs and 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, the TBUs used are 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 following 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, the `tbs_pvbuss_s[i]` and `tbm_pvbuss_m[i]` port pair represent a TBU *i*, or `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 `pvbuss_id_routed_m` that is used to transport ATC Invalidates to the PCIe-RC.

To reduce system construction complexity, the `tbs_pvbuss_s[i]` and `tbm_pvbuss_m[i]` pair also acts as a TBU so that the PCIe-RC does not need to 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 ATC Invalidates must be routed to the correct PCIe subsystem to invalidate the cache of ATS Responses in the subsystem. Therefore 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, 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.

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 set to one.

Debug and real accesses to the registers must be 32 bits or 64 bits.

MSIs are issued on the `qtw_pvbustm` port using attributes that are determined by the parameter `msi_attribute_transform`, while Event queue writes are always issued with `ExtendedID=0`, `UserFlags=0`, `MasterID=0xFFFFffff`. In the hardware, there is no way of distinguishing Event queue writes from MSI writes. However, this provides a mechanism for the model system to distinguish them.

The hardware only has a single cacheability attribute for input transactions, but `PVBus` transports have both inner and outer cacheability.

For non-PCIe-mode TBUs, whose 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, and it is Write-back, 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).

Therefore, 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 value. If it is iNC-oNC-osh, then it must be presented as such.

For PCIe-mode TBUs, that is, whose 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 is interpreted as iNC-oNC-osh. ! NoSnoop is 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, it is forced to iNC-oNC-osh. For example, if a NoSnoop transaction uses a page table and is transformed to iWB-oWB-nsh then it is forced to iNC-oNC-osh. However, if the page tables transformed it to a device type, then as all device types are stronger than iNC-oNC-osh, it exits the SMMU as the device type.
- In the model, transactions are classified by their incoming memory attributes as to whether they are NoSnoop or not and then are normalized appropriately:
 - iWB-oWB-any-shareability are interpreted as ! NoSnoop, therefore are normalized to iWB-oWB-ish.
 - Anything else is considered NoSnoop, and therefore is normalized to iNC-oNC-osh.
- Translated accesses also have the same interpretation to determine NoSnoop and how they are normalized. Therefore they could enter the system with different attributes to if they entered the SMMU as Untranslated Accesses and were translated by the normal translation process, before exiting downstream of the SMMU and entering the system

- 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, it 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), it is normalized to iWB-oWB with the incoming shareability.
- If the input is anything else, it is normalized to iNC-oNC-osh.

The model accepts full architectural attributes of two levels of cacheability and so has to decide how to interpret this in terms of the hardware. For transactions that are not cache maintenance operations, the model replicates the outer attribute into the inner attribute and then performs the normalization that the hardware does.

The hardware normalizes the architectural output attributes and outputs a single level of cacheability and a user flag (OC) specifying if the architectural attributes were cacheable in the outer cacheable domain. If the transaction is classified as a PCIe transaction, the NoSnoop transform previously described is applied. That is, if the original transaction was NoSnoop then any weaker memory type is strengthened to iNC-oNC-osh, and the following transform is applied:

```

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.



Note

The model deals with groups of transactions with the same attributes and a similar range of addresses. The mapping that is used is stored by the bus infrastructure and is used for subsequent sufficiently similar transactions without needing the intervention of the SMMU model, and therefore are not traced.



Note

The value of the `version` parameter for MMU_700 can be either `r0p0` or `r1p0`.

Model limitations

- The Performance Monitoring Unit (PMU) has limited functionality. It is intended for demonstration purposes only and does not implement all architecturally required events. It does not implement the `pmusnapshot_ack` or `pmusnapshot_req` interface.

- The model does not implement:
 - RAS.
 - Power control.
 - `sup_oas`, which controls the OAS of the SMMU. This is expected to be constant for a system.
 - The SYSCO interface.
 - The Low-Power Interface.
 - Any of the IMP DEF MPAM registers.
 - The following registers in MMU-700 r1p0:
 - `TCU_SYSDISC{2-17}`
 - `TBU_SYSDISC{4-14}`
- Cache maintenance operations cannot be inserted into the TBU ports of the SMMU.
- PVBUS has no representation of the cache stash operations, so they are not supported.
- `TCU_CFG.XLATE_SLOTS` is fixed at 512.
- `TCU_STATUS.GNT_XLATE_SLOTS` always reads as 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, the `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 is 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.

Ports for MMU_700

Table 4-450: 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. NOTE that the model does not support <code>tcu_sid[31:0]</code> which is the MSI DeviceID to send on <code>axi_stream_msi_m</code> . Instead the parameter <code>smmu_msi_device_id</code> is used. 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>tcu_reset_in</code> .
<code>axi_stream_msi_m</code>	PVBUS	Master	Manager port used for sending SMMU originated MSIs directly to the GIC
<code>clk_in</code>	ClockSignal	Slave	Clock signal (in RTL <code>aclk</code>) This is a clock time-base used by the TCU to spread some of its processing over time, if enabled by the <code>wait_*</code> parameters. The clock must always be connected.
<code>cmd_sync_irpt_ns</code>	Signal	Master	Pulsed interrupt output signal for non-secure <code>CMD_SYNC</code> having a completion signal of <code>SIG_IRQ</code> .

Name	Protocol	Type	Description
cmd_sync_irpt_s	Signal	Master	Pulsed interrupt output signal for secure CMD_SYNC having a completion signal of SIG_IRQ.
event_q_irpt_ns	Signal	Master	Pulsed interrupt output signal for the non-secure event queue becoming non-empty.
event_q_irpt_s	Signal	Master	Pulsed interrupt output signal for the secure event queue becoming non-empty.
evento	Signal	Master	Event signal
global_irpt_ns	Signal	Master	Pulsed interrupt output signal for non-secure SMMU_GERROR(N) signalling an error.
global_irpt_s	Signal	Master	Pulsed interrupt output signal for secure SMMU_S_GERROR(N) signalling an error.
identify	SMMUv3AEMIdentifyProtocol	Master	Map the transaction to the tuple (StreamID, SubStreamID, SubStreamIDValid, SSD) The StreamID that is produced by the implementation of this protocol is not the final StreamID. The final StreamID is produced by using the list_of_ns_sid_high_at_bitpos0/list_of_s_sid_high_at_bitpos0 parameter to map the StreamID based on the upstream port index. Also see the parameter howto_identify which can replace the functionality of this port under certain circumstances.
pri_q_irpt_ns	Signal	Master	Pulsed interrupt output signal for the PRI queue becoming non-empty.
prog_pvbus_s	PVBus	Slave	Register subordinate port (in RTL PROG)
pvbus_id_routed_m[62]	PVBus	Master	This is a special "id-routed" port for transmitting ATC invalidates upstream into the PCIe EndPoints, it is not a normal bus. 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 when separate_tw_msi_qs_port=true (in RTL QTW)
sec_override	Signal	Slave	Allow certain registers to be accessible to non-secure accesses from reset, as described in the TCU_SCR register.
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.
sup_cohacc	Signal	Slave	System supports COHACC and will be reflected in the IDR registers.
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.
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.
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.

Name	Protocol	Type	Description
tbs_pvbus_s[62]	PVBus	Slave	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered pvbus_m[] port.
tbu_cri_irpt[62]	Signal	Master	Critical error interrupt for RAS events from the TBU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE in the hardware then this is called ras_cri.
tbu_eri_irpt[62]	Signal	Master	Error Recovery Interrupt for RAS events from the TBU. This is used for uncorrected errors. NOTE in the hardware then this is called ras_eri.
tbu_fhi_irpt[62]	Signal	Master	Fault Handling Interrupt for RAS events from the TBU. Usually this is for corrected errors. NOTE in the hardware then this is called ras_fhi.
tbu_pmu_irpt[62]	Signal	Master	TBU Performance Monitoring Unit interrupt, one per TBU.
tbu_pmusnapshot_ack[62]	Signal	Master	PMU snapshot interface for the TBU, ack a snapshot.
tbu_pmusnapshot_req[62]	Signal	Slave	PMU snapshot interface for the TBU, request a snapshot.
tbu_reset_in[62]	Signal	Slave	Reset signals The TBUs can have independent reset signals. 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

all_error_messages_through_trace

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. .

Type: bool. Default value: 0x0.

axi_stream_msi_addr_to_match

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 that 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 .

Type: int. Default value: -0x1.

axi_stream_msi_TDEST

ID of the AXI stream subordinate port on the GIC that receives SMMU originated MSIs sent directly. The name of the GIC port is axi_stream_msi_s. NOTE that 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 .

Type: int. Default value: 0x0.

axi_stream_msi_TID

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 that 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 .

Type: int. Default value: 0x0.

behaviour_of_sampled_at_reset_signals

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. .

Type: int. Default value: 0x0.

cmdq_max_number_of_commands_to_buffer

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 will accentuate this effect. .

Type: int. Default value: 0xa.

enable_device_id_checks

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 then if it overflows 32 bits then the model will warn. If the DeviceID is not used then it is assumed that the external agent that forms the DeviceID will warn if it overflows. * If the SMMU supports MSIs, then the model will check that the GIC will be 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. .

Type: bool. Default value: 0x1.

howto_identify

If 'use-identify' then will use 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. The StreamID is secure if SEC_SID is true. The SubstreamID (20 b) is valid if SSV is true. NoStreamID transactions are identified by SIDV == 0 and the SSD is the PAS of the transaction and SEC_SID is not used. nSEC_SID, nSSV, nSIDV are available with negative logic. Negative and positive logic symbols for the same attribute is a error. Different attributes are independent and can use negative or positive logic. .

Type: string. Default value: "use-identify".

ish_is_osh_DANGER

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. NOTE that 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. .

Type: bool. Default value: 0x1.

list_of_ns_sid_high_at_bitpos0

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. .

Type: string. Default value: "".

list_of_pcie_mode

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 attributes handling for these TBUs are 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" .

Type: `string`. Default value: `""`.

list_of_pcie_rc

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, then the PCIe-RC uses this port for ATS/PRI, and then the actual transactions go through separate TBUs. In the model, then 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. .

Type: `string`. Default value: `""`.

list_of_s_sid_high_at_bitpos0

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. .

Type: `string`. Default value: `""`.

mpam_attribute_transform

If MPAM is supported, this is applied to `_all_` downstream transactions to transport the MPAM information. * `""/"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. .

Type: `string`. Default value: `"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS"`.

msi_attribute_transform

Transform downstream attributes of MSI transactions. * `""/"none"` -- no transform * How to alter output attributes of SMMU-generated MSIs. Example: "UserFlags[15:0]=smmu_msi_device_id[31:16], MasterID[15:0]=smmu_msi_device_id[15:0], ExtendedID=0" RHS/LHS Symbols: * ExtendedID/MasterID/UserFlags RHS Symbols: * the parameter `smmu_msi_device_id` * the symbol 'interrupt_kind' * 0/1 -- EVENTQ s/ns * 2 -- PRIQ * 3/4 -- CMD_SYNC s/ns * 5/6 -- GERRROR s/ns * 7/8 -- PMCG s/ns * 9/10/11 -- RAS FHI/ERI/CRI * HWATTR_KIND_0: PBHA information * numeric literals. ExtendedID/MasterID/UserFlags start with values {0, 0xFFFFfff, 0} respectively. Any bits with no transform are unchanged. This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices. NOTE: * see also 'output_attribute_transform' and `enable_device_id_checks`. .

Type: `string`. Default value: `"ExtendedID[31:0]=smmu_msi_device_id, MasterID=0xFFFFfff"`.

normalize_input_normal_non_iWB_oWB_to_iNC_oNC_osh_DANGER

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. NOTE that 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. .

Type: `bool`. Default value: `0x1`.

number_of_ports

The number of port pairs that the SMMU has. .

Type: `int`. Default value: `0x1`.

output_attribute_transform

Transform the downstream attributes of a translated transaction. * ""/"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. .

Type: `string`. Default value: "ExtendedID[31:0]=DeviceID".

output_id_routed_transform

The SMMU generates the following ID-routed transaction on the `pvbus_id_routed_m` bus:

- * ATC Invalidate
- * PRI Response This parameter expresses 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 .

Type: `string`. Default value: "Address[27:12]=StreamID[15:0], PAS=SSD".

prefetch_only_requests

The simulator supports 'prefetch-only' DMI requests, which can occur for anytime 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()`. .

Type: `int`. Default value: `0x0`.

sec_override

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. .

Type: `bool`. Default value: `0x0`.

seed

Used to seed the pseudo-random number generator that the SMMU model uses. .

Type: `int`. Default value: `0x12345678`.

size_of_cd_cache

The number of entries in the cache holding CD structures. 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. .

Type: `int`. Default value: `0x0`.

size_of_llcd_cache

The number of entries in the cache holding L1CD descriptors. 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. .

Type: `int`. Default value: `0x0`.

size_of_llste_cache

The number of entries in the cache holding L1STE descriptors. 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. .

Type: `int`. Default value: `0x0`.

size_of_ste_cache

The number of entries in the cache holding STE structures. 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. .

Type: `int`. Default value: `0x0`.

size_of_tlb

The number of entries in the 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. .

Type: `int`. Default value: `0x0`.

smmu_msi_device_id

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID. See parameter `msi_attribute_transform` and `enable_device_id_checks`. .

Type: `int`. Default value: `0x0`.

sup_cohacc

The value to put in `SMMU_IDR0.COHAAC` .

Type: `bool`. Default value: `0x1`.

sup_oas

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 .

Type: `int`. Default value: `0x6`.

TCUCFG_PARTID_WIDTH

The width of the MPAM PARTID on the bus. Valid values are 1, 6 and 9. See also parameter `mpam_attribute_transform`. .

Type: `int`. Default value: `0x9`.

TCUCFG_XLATE_SLOTS

Maximum number of outstanding stalled transactions that the SMMU supports Note that `TCUCFG_XLATE_SLOTS` must be \geq `TCUCFG_PTW_SLOTS` which currently fixed to 512 Accepted Values: 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 .

Type: `int`. Default value: `0x200`.

tlb_when_do_f_tlb_conflict_on_overlap

If a TLB entry is created by a walk and it overlaps an existing entry. Then 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. .

Type: `int`. Default value: `0x0`.

translated_device_id_base

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`. .

Type: `int`. Default value: `0x0`.

tw_qs_attribute_transform

Transform downstream attributes of table walk and queue transactions. *

""/"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 * numeric literals.

ExtendedID/MasterID/UserFlags start with values {0, 0xFFFFFFFF, 0} respectively. Any bits with no transform are unchanged. NOTE: * see also 'output_attribute_transform' and 'msi_attribute_transform'. .

Type: `string`. Default value: `""`.

version

The version of this product. Valid values are: `r0p0` `r1p0` .

Type: `string`. Default value: `"r0p0"`.

wait_cmdq_ticks

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))`]. .

Type: `int`. Default value: `0x0`.

wait_eventq_ticks

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 $[0, \max(2, (t \& 0xFFFFfff))]$. .

Type: `int`. Default value: `0x0`.

wait_misc_async_actions_ticks

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 $[0, \max(2, (t \& 0xFFFFfff))]$. .

Type: `int`. Default value: `0x0`.

wait_msi_ticks

This is the time to wait before sending an MSI. If bit 32 is set (`0x1_0000_0000`) then the time waited for is a uniform randomly distributed time $[0, \max(2, (t \& 0xFFFFfff))]$. .

Type: `int`. Default value: `0x0`.

wait_pri_req_ticks

This is the time to wait before processing a PRI Request. If bit 32 is set (`0x1_0000_0000`) then the time waited for is a uniform randomly distributed time $[0, \max(2, (t \& 0xFFFFfff))]$. .

Type: `int`. Default value: `0x0`.

wait_pri_resp_ticks

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, that immediately makes a PRI Request, that auto-responds, etc. To break this loop, then we introduce a minimum time on all PRI Responses to give other components in the system a chance to run. If bit 32 is set (`0x1_0000_0000`) then the time waited for is a uniform randomly distributed time $[0, \max(2, (t \& 0xFFFFfff))]$. .

Type: `int`. Default value: `0x1`.

4.10.51 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 4-451: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

MemoryMappedCounterModule contains the following CADI targets:

- MemoryMappedCounterModule

MemoryMappedCounterModule contains the following MTI components:

- [MemoryMappedCounterModule](#)
- [PVBusSlave](#)

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 other than the input clock to the core.



The component has two bus slave ports because the architecture specification permits you to map each set of registers at different, non-contiguous base addresses.

Ports for MemoryMappedCounterModule

Table 4-452: 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

Reset value for CNTFID0, base frequency in Hz.

Type: `int`. Default value: `0x5f5e100`.

cntcidr0123_C

Values to be returned for control-frame CIDR registers.

Type: `int`. Default value: `0x0`.

cntcidr0123_R

Values to be returned for read-frame CIDR registers.

Type: `int`. Default value: `0x0`.

cntpidr0123_C

Values to be returned for control-frame PIDR registers 0-3.

Type: `int`. Default value: `0x0`.

cntpidr0123_R

Values to be returned for read-frame PIDR registers 0-3.

Type: `int`. Default value: `0x0`.

cntpidr4567_C

Values to be returned for control-frame PIDR registers 4-7.

Type: `int`. Default value: `0x0`.

cntpidr4567_R

Values to be returned for read-frame PIDR registers 4-7.

Type: `int`. Default value: `0x0`.

diagnostics

Diagnostics.

Type: `int`. Default value: `0x0`.

has_counter_scaling

Implements ARMv8.4 generic counter scaling (FEAT_CNTSC).

Type: `bool`. Default value: `0x0`.

non_arch_fixed_frequency

If set, ignore CNTFID0 and instead use this frequency in Hz.

Type: `int`. Default value: `0x0`.

non_arch_start_at_default

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.

Type: `bool`. Default value: `0x0`.

readonly_is_WI

Ignore (rather than failing) on writes to read-frame.

Type: `bool`. Default value: `0x0`.

use_real_time

****Deprecated, this parameter will be removed in future versions**** Update the Generic Timer counter at a real-time base frequency instead of simulator time.

Type: `bool`. Default value: `0x0`.

4.10.52 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 4-453: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

MessageHandlingUnitV2 contains the following CADI targets:

- MessageHandlingUnitV2

MessageHandlingUnitV2 contains the following MTI components:

- [MessageHandlingUnitV2](#)

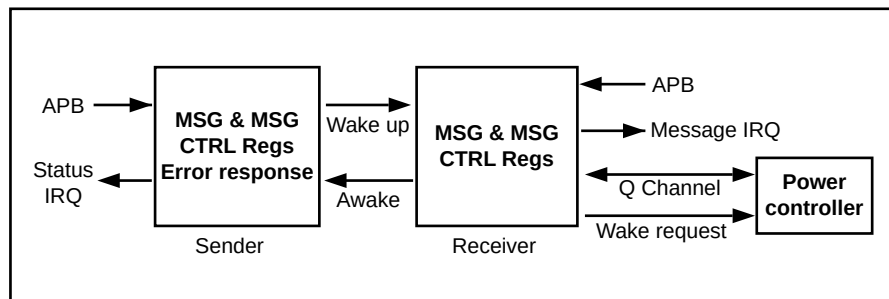
About MessageHandlingUnitV2

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 4-5: MessageHandlingUnitV2 structure



Ports for MessageHandlingUnitV2

Table 4-454: Ports

Name	Protocol	Type	Description
int_access_nr2r	Signal	Master	-
int_access_r2nr	Signal	Master	-
mhu_combined_irq	Signal	Master	-
mhu_irq[124]	Signal	Master	-
mhu_snd_irq[124]	Signal	Master	-
pvbus_s_rec	PVBus	Slave	-
pvbus_s_snd	PVBus	Slave	-
qchannel_mhu_pwr	PChannel	Slave	-
reset_rec	Signal	Slave	-
reset_snd	Signal	Slave	-
snd_combined_irq	Signal	Master	-
wakerequest	Signal	Master	-

4.10.53 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 4-455: IP revisions support

Revision	Quality level
r0p0	Preliminary support
r1p0	Preliminary support
r2p0	Preliminary support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

NI700 contains the following CADI targets:

- NI700

NI700 contains the following MTI components:

- [NI700](#)
- [PVBusMapper](#)

Configuring and using the model

- For information about the changes in this release, see the Fast Models Portfolio release notes.
- 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 IYML 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](#).
- The mapping between the port number for ASNI, AMNI, HSNI, HMNI, and PMNI ports and the xxNI name used in the yml file is its index when all xxNIs of that same type are sorted in ascending alphabetical order. For example, if the yml file has three ASNIs `asni_s100_scp`, `asni_s101_dap`, and `asni_s204_periph0`:
 - `asni_s100_scp` is mapped to `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_hsni`
- 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.
- 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 4-456: Ports

Name	Protocol	Type	Description
<code>idm_reset_signal_amni[127]</code>	Signal	Master	IDM reset signals to AMNIs.
<code>idm_reset_signal_asni[128]</code>	Signal	Master	IDM reset signals to ASNIs.
<code>idm_reset_signal_hmni[127]</code>	Signal	Master	IDM reset signals to HMNIs.
<code>idm_reset_signal_hsni[128]</code>	Signal	Master	IDM reset signals to HSNIs.
<code>idm_reset_signal_pmni[127]</code>	Signal	Master	IDM reset signals to PMNIs.
<code>pvbus_m_amni[127]</code>	PVBus	Master	AMNI downstream ports.
<code>pvbus_m_hmni[127]</code>	PVBus	Master	HMNI downstream ports.
<code>pvbus_m_pmni[127]</code>	PVBus	Master	PMNI downstream ports.
<code>pvbus_s_asni[128]</code>	PVBus	Slave	ASNI upstream ports.
<code>pvbus_s_hsni[128]</code>	PVBus	Slave	HSNI upstream ports.

Name	Protocol	Type	Description
reset_in	Signal	Slave	Reset signal.

Parameters for NI700

ipxact_config_file

null.

Type: string. Default value: "".

mesh_config_file

Name of a file containing mesh placement of NI700 components.

Type: string. Default value: "".

mpam_attributes

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.

Type: string. Default value: "".

periphbase

Value for PERIPHBASE.

Type: int. Default value: -0x1.

print_config

Enables printing the config register addresses.

Type: bool. Default value: 0x0.

print_parser_log

Enables printing the yaml config parser log messages.

Type: bool. Default value: 0x0.

revision

Component revision. Currently supports r0p0, r1p0, r2p0.

Type: string. Default value: "r2p0".

show_banner

Show component banner: 0=>supress entire banner, 1=>suppress config file, 2+=>show full banner.

Type: int. Default value: 0x2.

4.10.54 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 4-457: IP revisions support

Revision	Quality level
r1p4	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL011_Uart contains the following CADI targets:

- ClockDivider
- ClockTimerThread
- ClockTimerThread64
- PL011_Uart
- SchedulerThread
- SchedulerThreadEvent

PL011_Uart contains the following MTI components:

- [ClockDivider](#)
- [PL011_Uart](#)
- [PVBusSlave](#)

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
```



A different escape sequence can be set in place of `##` using the parameter `in_file_escape_sequence`.

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.

Differences between the model and the RTL

This component does not implement the DMA functionality of the PL011 PrimeCell.

Ports for PL011_Uart

Table 4-458: Ports

Name	Protocol	Type	Description
clk_in_ref	ClockSignal	Slave	Clock input, typically 14.745MHz, which sets the master transmit/receive rate.
intr	Signal	Master	Interrupt signal.
pvbuss	PVBus	Slave	Slave port for register access.
serial_out	SerialData	Master	Serial input/output and control signals. Used to communicate with a serial device, such as a terminal.

Parameters for PL011_Uart

baud_rate

Baud rate.

Type: `int`. Default value: `0x9600`.

clock_rate

Clock rate for PL011.

Type: `int`. Default value: `0xe10000`.

enable_dc4

Enable DC4 commands (try `echo -e "help\024"` in a Linux shell in a serial console).

Type: `bool`. Default value: `0x1`.

flow_ctrl_mask_en

Enable hardware flow control workaround which forcefully disables CTSen and RTSen bits in UARTCR register.

Type: `bool`. Default value: `0x0`.

halt

Halt instead of shutdown for `shutdown_on_eot` and `shutdown_tag`.

Type: `bool`. Default value: `0x0`.

in_file

Input file for data to be read by the UART.

Type: `string`. Default value: `""`.

in_file_escape_sequence

Input file escape sequence.

Type: `string`. Default value: `""`.

out_file

Output file to hold data written by the UART (use '-' to send all output to stdout).

Type: `string`. Default value: `""`.

revision

Revision to simulate.

Type: `string`. Default value: `"r1p4"`.

shutdown_on_eot

Shutdown simulation when a EOT (ASCII 4) char is transmitted (useful for regression tests when semihosting is not available).

Type: `bool`. Default value: `0x0`.

shutdown_tag

Shutdown simulation when a string is transmitted.
Type: `string`. Default value: `""`.

toggle_mti

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.
Type: `string`. Default value: `""`.

uart_enable

Enable uart when the system starts up. (clock_rate and baud_rate are only valid when this option is enabled).
Type: `bool`. Default value: `0x0`.

unbuffered_output

Unbuffered output.
Type: `bool`. Default value: `0x0`.

untimed_fifos

Ignore the clock rate and transmit/receive serial data immediately.
Type: `bool`. Default value: `0x1`.

Related information

[TelnetTerminal](#) on page 1169

4.10.55 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 4-459: IP revisions support

Revision	Quality level
r1p4	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL022_SSP contains the following CADI targets:

- ClockDivider
- PL022_SSP

PL022_SSP contains the following MTI components:

- [ClockDivider](#)

- [PVBusSlave](#)

Differences between the model and the RTL

Although the PL022_SSP component has clock input, it is not internally clock-driven. This is different to the hardware.



This component is a preliminary release. It is not a fully-supported peripheral.

Ports for PL022_SSP

Table 4-460: 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.

4.10.56 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 4-461: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL030_RTC contains the following CADI targets:

- ClockTimerThread

- ClockTimerThread64
- PL030_RTC
- SchedulerThread
- SchedulerThreadEvent

PL030_RTC contains the following MTI components:

- [PVBusSlave](#)

Ports for PL030_RTC

Table 4-462: 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.

4.10.57 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 4-463: IP revisions support

Revision	Quality level
r1p3	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL031_RTC contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL031_RTC
- SchedulerThread
- SchedulerThreadEvent

PL031_RTC contains the following MTI components:

- [PVBusSlave](#)

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 4-464: Ports

Name	Protocol	Type	Description
clock	ClockSignal	Slave	Clock input, typically 1MHz, driving master count rate.
intr	Signal	Master	Interrupt signaling.
pvbust	PVBus	Slave	Slave port for connection to PV bus master/decoder.

Parameters for PL031_RTC

RTCDR_reset_value

Reset value for RTCDR.

Type: int. Default value: 0x0.

RTCDR_use_current_time

Use current Unix/POSIX time for reset value for RTCDR. If true RTCDR_reset_value is ignored.

Type: bool. Default value: 0x1.

4.10.58 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 4-465: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL041_AACI contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL041_AACI
- SchedulerThread
- SchedulerThreadEvent

PL041_AACI contains the following MTI components:

- [PVBusSlave](#)

About PL041_AACI

The PL041_AACI component is designed to connect to an audio output component such as AudioOutFile 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 4-466: 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.
pvbuss	PVBuss	Slave	Slave port for connection to PV bus master/decoder.

Parameters for PL041_AACI

enabled

Host interface connection enabled.

Type: bool. Default value: 0x1.

Related information

[AudioOut_File](#) on page 1139

[AudioOut_SDL](#) on page 1140

4.10.59 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 4-467: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL050_KMI contains the following CADI targets:

- ClockDivider
- PL050_KMI

PL050_KMI contains the following MTI components:

- [ClockDivider](#)
- [PVBusSlave](#)

About PL050_KMI

This model communicates with models of PS/2-like devices, for example a PS2Keyboard or PS2Mouse.

Ports for PL050_KMI

Table 4-468: 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.

4.10.60 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 4-469: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL061_GPIO contains the following CADI targets:

- PL061_GPIO

PL061_GPIO contains the following MTI components:

- [PVBusSlave](#)

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 4-470: Ports

Name	Protocol	Type	Description
GPIO_In	Value	Slave	Input lines. 32-bit data in, only [7:0] is used.
GPIO_Intr	Signal	Master	Interrupt signal indicating to an interrupt controller that an interrupt occurred in one or more of the GPIO_In lines.
GPIO_MIS	Value	Master	Indicates the masked interrupt status. 32-bit data out , only [7:0] is used. NOT necessary, as the GPIOMIS can be read from address 0x418.
GPIO_Out	Value	Master	Output lines. 32-bit data out, only [7:0] is used.
pvbuss	PVBus	Slave	Slave port for register access.

Parameters for PL061_GPIO

init_inputs

Default input values [7:0].

Type: `int`. Default value: `0x0`.

4.10.61 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 4-471: IP revisions support

Revision	Quality level
r1p3	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL080_DMAC contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL080_DMAC
- SchedulerThread
- SchedulerThreadEvent

PL080_DMAC contains the following MTI components:

- [PVBusMaster](#)

- [PVBusSlave](#)

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 4-472: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock signal to control DMA transfer rate.
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

request delay.

Type: int. Default value: 0x0.

fifo_size

Channel FIFO size in bytes.

Type: int. Default value: 0x10.

generate_clear

Generate clear response.

Type: bool. Default value: 0x0.

max_transfer

Largest atomic transfer.

Type: int. Default value: 0x100.

4.10.62 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 4-473: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL110_CLCD contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL110_CLCD
- PL11x_CLCD
- SchedulerThread
- SchedulerThreadEvent

PL110_CLCD contains the following MTI components:

- [PL11x_CLCD](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

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 4-474: 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.

Name	Protocol	Type	Description
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 PL110_CLCD

disable_snooping_dma

Disable DMA snooping.

Type: `bool`. Default value: `0x0`.

pixel_double_limit

Minimum LCD pixel width before display will be zoomed.

Type: `int`. Default value: `0x12c`.

Parameters for PL11x_CLCD

pl11x_clcd.disable_snooping_dma

Disable DMA snooping.

Type: `bool`. Default value: `0x0`.

pl11x_clcd.pixel_double_limit

Minimum LCD pixel width before display will be zoomed.

Type: `int`. Default value: `0x12c`.

pl11x_clcd.pl11x_behavior

Define PL11x behaviour. 0 for PL110, 1 for PL111.

Type: `int`. Default value: `0x0`.

Related information

[Visualisation library](#) on page 65

4.10.63 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 4-475: IP revisions support

Revision	Quality level
r0p2	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL111_CLCD contains the following CADI targets:

- ClockTimerThread

- ClockTimerThread64
- PL111_CLCD
- PL11x_CLCD
- SchedulerThread
- SchedulerThreadEvent

PL111_CLCD contains the following MTI components:

- [PL11x_CLCD](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

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 4-476: 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.
pvb主	PVBus	Slave	Slave port for register access.
pvb主_m	PVBus	Master	DMA port for video data.

Parameters for PL111_CLCD

disable_snooping_dma

Disable DMA snooping.
Type: bool. Default value: 0x0.

pixel_double_limit

Minimum LCD pixel width before display will be zoomed.
Type: int. Default value: 0x12c.

Parameters for PL11x_CLCD

pl11x_clcd.disable_snooping_dma

Disable DMA snooping.
Type: bool. Default value: 0x0.

pl11x_clcd.pixel_double_limit

Minimum LCD pixel width before display will be zoomed.
Type: int. Default value: 0x12c.

pl11x_clcd.pl11x_behavior

Define PL11x behaviour. 0 for PL110, 1 for PL111.

Type: `int`. Default value: `0x0`.

Related information

[Visualisation library](#) on page 65

4.10.64 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 4-477: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL180_MCI contains the following CADI targets:

- PL180_MCI

PL180_MCI contains the following MTI components:

- [PVBusSlave](#)

About PL180_MCI

When paired with an MMC card model, the PL180_MCI component provides emulation of a flexible, persistent storage mechanism. 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 4-478: Ports

Name	Protocol	Type	Description
MCI_INTR[2]	Signal	Master	Interrupts.
mmc_m	MMC_Protocol	Master	The MultiMediaCard (MMC) master port.
pvbuse	PVBus	Slave	Slave port for register access.

Parameters for PL180_MCI

p1180_fifo_depth

PL180 FIFO Depth.

Type: int. Default value: 0x10.

Related information

[MMC](#) on page 1463

4.10.65 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 4-479: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL192_VIC contains the following CADI targets:

- PL192_VIC

PL192_VIC contains the following MTI components:

- [PVBusSlave](#)

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.

Ports for PL192_VIC

Table 4-480: 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.
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.

4.10.66 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 4-481: IP revisions support

Revision	Quality level
rOp0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL310_L2CC contains the following CADI targets:

- PL310_L2CC

PL310_L2CC contains the following MTI components:

- [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 16KB to 8MB, 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 with use of 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 to 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 is 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 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.
- Specific Cortex®-A9 related optimizations are not supported: Prefetch hints, Full line of zero and Early write response.
- Hazard detection is not required because of the atomic nature of the accesses at 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 (like 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 to the model.

Ports for PL310_L2CC

Table 4-482: 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

Associativity for Auxiliary Control Register.

Type: `int`. Default value: `0x0`.

cache-state_modelled

Specifies whether real cache state is modelled (vs. register model).

Type: `bool`. Default value: `0x0`.

CACHEID

Cache controller cache ID.

Type: `int`. Default value: `0x0`.

CFGBIGEND

Big-endian mode for accessing configuration registers out of reset.

Type: `int`. Default value: `0x0`.

delay_cache_hit

Cost to handle a cache hit.

Type: `int`. Default value: `0x0`.

delay_cache_miss

Cost to handle a cache miss.
Type: `int`. Default value: `0x0`.

delay_cache_perbeat

Cost to handle one beat of cache data movement.
Type: `int`. Default value: `0x0`.

LOCKDOWN_BY_LINE

Lockdown by line - value is reflected in CacheType register Bit 25, but the feature is not switched off when the parameter is 0.
Type: `int`. Default value: `0x0`.

LOCKDOWN_BY_MASTER

Lockdown by master - value is reflected in CacheType register Bit 26, but the feature is not switched off when the parameter is 0.
Type: `int`. Default value: `0x0`.

REGFILEBASE

Base address for accessing configuration registers.
Type: `int`. Default value: `0x1f002000`.

WAYSIZE

Size of ways for Auxiliary Control Register.
Type: `int`. Default value: `0x1`.

Related information

[Caches in PV models](#) on page 35

4.10.67 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 4-483: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL330_DMAC contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL330_DMAC
- SchedulerThread

- SchedulerThreadEvent

PL330_DMAC contains the following MTI components:

- [PL330_DMAC](#)
- [PVBUSMaster](#)
- [PVBUSSlave](#)

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, the ID is AxID[3:0], with $0 \times 0 - (\text{numberOfChannels} - 1)$ identifying channels and $(\text{numberOfChannels})$ identifying the manager. For example, $0 \times 0 - 0 \times 7$ and 0×8 , 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: $0 \times \text{ffff}$.
- Manager instruction fetch: $0 \times \text{ffff}$.

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 4-484: 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.

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	Slave port for all register accesses (secure).
pvbus_s_ns	PVBus	Slave	Slave port for all register accesses (non-secure).
reset_in	Signal	Slave	System reset.

Parameters for PL330_DMAC

activate_delay

request delay.

Type: int. Default value: 0x0.

fifo_size

Channel FIFO size in bytes.

Type: int. Default value: 0x10.

generate_clear

Generate clear response.

Type: bool. Default value: 0x0.

max_transfer

Largest atomic transfer.

Type: int. Default value: 0x100.

p_axi_bus_width_param

AXI bus width.

Type: int. Default value: 0x20.

p_buffer_depth

buffer depth.

Type: int. Default value: 0x10.

p_cache_line_words

number of words in a cache line.

Type: int. Default value: 0x1.

p_cache_lines

number of cache lines.

Type: int. Default value: 0x1.

p_controller_boots

DMA boots from reset.

Type: bool. Default value: 0x1.

p_controller_nsecure

Controller non-secure at reset (boot_manager_ns).

Type: bool. Default value: 0x0.

p_irq_nsecure

Interrupts non-secure at reset.

Type: int. Default value: 0x0.

p_lsq_read_size

LSQ read buffer depth.

Type: `int`. Default value: `0x4`.**p_lsq_write_size**

LSQ write buffer depth.

Type: `int`. Default value: `0x4`.**p_max_channels**

virtual channels.

Type: `int`. Default value: `0x8`.**p_max_irqs**

number of interrupts.

Type: `int`. Default value: `0x20`.**p_max_periph**

number of peripheral interfaces.

Type: `int`. Default value: `0x20`.**p_perip_request_acceptance_0**

Peripheral 0 request acceptance.

Type: `int`. Default value: `0x2`.**p_perip_request_acceptance_1**

Peripheral 1 request acceptance.

Type: `int`. Default value: `0x2`.**p_perip_request_acceptance_10**

Peripheral 10 request acceptance.

Type: `int`. Default value: `0x2`.**p_perip_request_acceptance_11**

Peripheral 11 request acceptance.

Type: `int`. Default value: `0x2`.**p_perip_request_acceptance_12**

Peripheral 12 request acceptance.

Type: `int`. Default value: `0x2`.**p_perip_request_acceptance_13**

Peripheral 13 request acceptance.

Type: `int`. Default value: `0x2`.**p_perip_request_acceptance_14**

Peripheral 14 request acceptance.

Type: `int`. Default value: `0x2`.**p_perip_request_acceptance_15**

Peripheral 15 request acceptance.

Type: `int`. Default value: `0x2`.

- p_perip_request_acceptance_16**
Peripheral 16 request acceptance.
Type: `int`. Default value: `0x2`.
- p_perip_request_acceptance_17**
Peripheral 17 request acceptance.
Type: `int`. Default value: `0x2`.
- p_perip_request_acceptance_18**
Peripheral 18 request acceptance.
Type: `int`. Default value: `0x2`.
- p_perip_request_acceptance_19**
Peripheral 19 request acceptance.
Type: `int`. Default value: `0x2`.
- p_perip_request_acceptance_2**
Peripheral 2 request acceptance.
Type: `int`. Default value: `0x2`.
- p_perip_request_acceptance_20**
Peripheral 20 request acceptance.
Type: `int`. Default value: `0x2`.
- p_perip_request_acceptance_21**
Peripheral 21 request acceptance.
Type: `int`. Default value: `0x2`.
- p_perip_request_acceptance_22**
Peripheral 22 request acceptance.
Type: `int`. Default value: `0x2`.
- p_perip_request_acceptance_23**
Peripheral 23 request acceptance.
Type: `int`. Default value: `0x2`.
- p_perip_request_acceptance_24**
Peripheral 24 request acceptance.
Type: `int`. Default value: `0x2`.
- p_perip_request_acceptance_25**
Peripheral 25 request acceptance.
Type: `int`. Default value: `0x2`.
- p_perip_request_acceptance_26**
Peripheral 26 request acceptance.
Type: `int`. Default value: `0x2`.
- p_perip_request_acceptance_27**
Peripheral 27 request acceptance.
Type: `int`. Default value: `0x2`.

p_perip_request_acceptance_28

Peripheral 28 request acceptance.
Type: `int`. Default value: `0x2`.

p_perip_request_acceptance_29

Peripheral 29 request acceptance.
Type: `int`. Default value: `0x2`.

p_perip_request_acceptance_3

Peripheral 3 request acceptance.
Type: `int`. Default value: `0x2`.

p_perip_request_acceptance_30

Peripheral 30 request acceptance.
Type: `int`. Default value: `0x2`.

p_perip_request_acceptance_31

Peripheral 31 request acceptance.
Type: `int`. Default value: `0x2`.

p_perip_request_acceptance_4

Peripheral 4 request acceptance.
Type: `int`. Default value: `0x2`.

p_perip_request_acceptance_5

Peripheral 5 request acceptance.
Type: `int`. Default value: `0x2`.

p_perip_request_acceptance_6

Peripheral 6 request acceptance.
Type: `int`. Default value: `0x2`.

p_perip_request_acceptance_7

Peripheral 7 request acceptance.
Type: `int`. Default value: `0x2`.

p_perip_request_acceptance_8

Peripheral 8 request acceptance.
Type: `int`. Default value: `0x2`.

p_perip_request_acceptance_9

Peripheral 9 request acceptance.
Type: `int`. Default value: `0x2`.

p_periph_nsecure

Peripherals non-secure at reset.
Type: `bool`. Default value: `0x0`.

p_read_issuing_capability

AXI read issuing capability.
Type: `int`. Default value: `0x1`.

p_reset_pc

DMA PC at reset.

Type: int. Default value: 0x60000000.

p_write_issuing_capability

AXI write issuing capability.

Type: int. Default value: 0x1.

revision

revision ID.

Type: string. Default value: "r0p0".

4.10.68 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 4-485: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL340_DMC contains the following CADI targets:

- PL340_DMC
- PVBusExclusiveMonitor

PL340_DMC contains the following MTI components:

- [PVBusExclusiveMonitor](#)
- [PVBusMapper](#)
- [PVBusSlave](#)

PL340_DMC - about

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 4-486: 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

Set this parameter to 0 if memory is connected.
Type: `int`. Default value: `-0x1`.

IF_CHIP1

Set this parameter to 0 if memory is connected.
Type: `int`. Default value: `-0x1`.

IF_CHIP2

Set this parameter to 0 if memory is connected.
Type: `int`. Default value: `-0x1`.

IF_CHIP3

Set this parameter to 0 if memory is connected.
Type: `int`. Default value: `-0x1`.

MEMORY_WIDTH

Set this parameter to 0 if memory is connected.
Type: `int`. Default value: `0x20`.

Parameters for PVBusExclusiveMonitor

exclusive_monitor0.apply_access_width_criteria_to_non_excl_stores

Apply the given exclusive store width matching criteria to non-exclusive stores.
Type: `bool`. Default value: `0x1`.

exclusive_monitor0.clear_on_strex_address_mismatch

Whether monitor is cleared when strex fails due to address mismatch.
Type: `bool`. Default value: `0x1`.

exclusive_monitor0.enable_component

Enable component.
Type: `bool`. Default value: `0x1`.

exclusive_monitor0.exclusive_monitor_clear_on_atomic_from_same_master

Monitor atomics from the same master.
Type: `bool`. Default value: `0x1`.

exclusive_monitor0.log2_granule_size

log2 of address granule size.
Type: `int`. Default value: `0x0`.

exclusive_monitor0.match_access_width

Fail STREX if not the same access width as LDREX.
Type: `bool`. Default value: `0x0`.

exclusive_monitor0.match_secure_state

Treat the secure state like an address bit.
Type: `bool`. Default value: `0x1`.

exclusive_monitor0.monitor_access_level

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

Type: `int`. Default value: `0x0`.

exclusive_monitor0.monitor_non_excl_stores

Monitor non-exclusive stores from the same master.

Type: `bool`. Default value: `0x0`.

exclusive_monitor0.number_of_monitors

Number of monitors.

Type: `int`. Default value: `0x8`.

exclusive_monitor0.shareability_domain

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

Type: `int`. Default value: `0x3`.

4.10.69 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 4-487: IP revisions support

Revision	Quality level
r1p2	Full support
r2p2	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL350_SMC contains the following CADI targets:

- PL350_SMC
- PVBusExclusiveMonitor
- TZSwitch

PL350_SMC contains the following MTI components:

- [PVBusExclusiveMonitor](#)
- [PVBusMapper](#)
- [PVBusMaster](#)
- [PVBusSlave](#)

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 4-488: 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_CHIPO

Interface 0 chip 0 connected.
Type: bool. Default value: 0x0.

IF0_CHIPO_BASE

Interface 0 chip 0 Base address.
Type: int. Default value: 0x0.

IF0_CHIPO_SIZE

Interface 0 chip 0 Size.
Type: int. Default value: 0x0.

IF0_CHIP1

Interface 0 chip 1 connected.
Type: bool. Default value: 0x0.

IF0_CHIP1_BASE

Interface 0 chip 1 Base address.
Type: int. Default value: 0x0.

IF0_CHIP1_SIZE

Interface 0 chip 1 Size.

Type: `int`. Default value: `0x0`.

IF0_CHIP2

Interface 0 chip 2 connected.

Type: `bool`. Default value: `0x0`.

IF0_CHIP2_BASE

Interface 0 chip 2 Base address.

Type: `int`. Default value: `0x0`.

IF0_CHIP2_SIZE

Interface 0 chip 2 Size.

Type: `int`. Default value: `0x0`.

IF0_CHIP3

Interface 0 chip 3 connected.

Type: `bool`. Default value: `0x0`.

IF0_CHIP3_BASE

Interface 0 chip 3 Base address.

Type: `int`. Default value: `0x0`.

IF0_CHIP3_SIZE

Interface 0 chip 3 Size.

Type: `int`. Default value: `0x0`.

IF0_MEM_TYPE_PARAMETER

Interface 0 Mem type.

Type: `int`. Default value: `0x0`.

IF1_CHIP0

Interface 1 chip 0 connected.

Type: `bool`. Default value: `0x0`.

IF1_CHIP0_BASE

Interface 1 chip 0 Base address.

Type: `int`. Default value: `0x0`.

IF1_CHIP0_SIZE

Interface 1 chip 0 Size.

Type: `int`. Default value: `0x0`.

IF1_CHIP1

Interface 1 chip 1 connected.

Type: `bool`. Default value: `0x0`.

IF1_CHIP1_BASE

Interface 1 chip 1 Base address.

Type: `int`. Default value: `0x0`.

IF1_CHIP1_SIZE

Interface 1 chip 1 Size.

Type: `int`. Default value: `0x0`.**IF1_CHIP2**

Interface 1 chip 2 connected.

Type: `bool`. Default value: `0x0`.**IF1_CHIP2_BASE**

Interface 1 chip 2 Base address.

Type: `int`. Default value: `0x0`.**IF1_CHIP2_SIZE**

Interface 1 chip 2 Size.

Type: `int`. Default value: `0x0`.**IF1_CHIP3**

Interface 1 chip 3 connected.

Type: `bool`. Default value: `0x0`.**IF1_CHIP3_BASE**

Interface 1 chip 3 Base address.

Type: `int`. Default value: `0x0`.**IF1_CHIP3_SIZE**

Interface 1 chip 3 Size.

Type: `int`. Default value: `0x0`.**IF1_MEM_TYPE_PARAMETER**

Interface 1 Mem type.

Type: `int`. Default value: `0x0`.**PERIPH_ID_0**

Periph_ID_0 value.

Type: `int`. Default value: `0x52`.**REMAP**

Remap the device.

Type: `int`. Default value: `-0x1`.**revision**

Revision.

Type: `string`. Default value: `"r1p2"`.**Parameters for PVBusExclusiveMonitor****exclusive_monitor0.apply_access_width_criteria_to_non_excl_stores**

Apply the given exclusive store width matching criteria to non-exclusive stores.

Type: `bool`. Default value: `0x1`.**exclusive_monitor0.clear_on_strex_address_mismatch**

Whether monitor is cleared when strex fails due to address mismatch.

Type: `bool`. Default value: `0x1`.

exclusive_monitor0.enable_component

Enable component.

Type: `bool`. Default value: `0x1`.

exclusive_monitor0.exclusive_monitor_clear_on_atomic_from_same_master

Monitor atomics from the same master.

Type: `bool`. Default value: `0x1`.

exclusive_monitor0.log2_granule_size

log2 of address granule size.

Type: `int`. Default value: `0x0`.

exclusive_monitor0.match_access_width

Fail STREX if not the same access width as LDREX.

Type: `bool`. Default value: `0x0`.

exclusive_monitor0.match_secure_state

Treat the secure state like an address bit.

Type: `bool`. Default value: `0x1`.

exclusive_monitor0.monitor_access_level

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

Type: `int`. Default value: `0x0`.

exclusive_monitor0.monitor_non_excl_stores

Monitor non-exclusive stores from the same master.

Type: `bool`. Default value: `0x0`.

exclusive_monitor0.number_of_monitors

Number of monitors.

Type: `int`. Default value: `0x8`.

exclusive_monitor0.shareability_domain

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

Type: `int`. Default value: `0x3`.

Parameters for TZSwitch

tzswitch_0.normal

Normal Port.

Type: `int`. Default value: `0x2`.

tzswitch_0.secure

Secure Port.

Type: `int`. Default value: `0x1`.

4.10.70 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 4-489: IP revisions support

Revision	Quality level
r1p2	Full support
r2p2	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL350_SMC_NAND_FLASH contains the following CADI targets:

- PL350_SMC_NAND_FLASH

PL350_SMC_NAND_FLASH contains the following MTI components:

- [PVBusSlave](#)

About PL350_SMC_NAND_FLASH

Program the component as you would the hardware.

Ports for PL350_SMC_NAND_FLASH

Table 4-490: Ports

Name	Protocol	Type	Description
irq	Signal	Master	Interrupt signaling.
pvbus	PVBus	Slave	Slave port for connection to PV bus master/decoder.

Parameters for PL350_SMC_NAND_FLASH

DEVICE_1

Device manufacturer code.

Type: `int`. Default value: `0xec`.

DEVICE_2

Device code.

Type: `int`. Default value: `0xda`.

DEVICE_3

Device 3rd cycle code.

Type: `int`. Default value: `0x80`.

DEVICE_4

Device 4th cycle code.

Type: `int`. Default value: `0x15`.

DEVICE_NAME

Device Name.

Type: string. Default value: "Samsung K9F1G08U0M".

NAND_BLOCK_COUNT
number of blocks in the flash device.
Type: int. Default value: 0x800.

NAND_FLASH_SIZE
flash size in byte.
Type: int. Default value: 0x10800000.

NAND_PAGE_COUNT_PER_BLOCK
number of pages in each block.
Type: int. Default value: 0x40.

NAND_PAGE_SIZE
page size.
Type: int. Default value: 0x840.

NAND_SPARE_SIZE_PER_PAGE
Spare size per page.
Type: int. Default value: 0x40.

NAND_VALID_SIZE_PER_PAGE
valid page size.
Type: int. Default value: 0x800.

4.10.71 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 4-491: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL370_HDLCD contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- PL370_HDLCD
- SchedulerThread
- SchedulerThreadEvent

PL370_HDLCD contains the following MTI components:

- [PVBusMaster](#)
- [PVBusSlave](#)



Too fast a pixel clock can slow the rest of the simulation.

Ports for PL370_HDLCD

Table 4-492: 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.
pvbus_m	PVBus	Master	DMA port for collecting video data from memory/framebuffer.

Parameters for PL370_HDLCD

diagnostics

Diagnostics level.

Type: `int`. Default value: `0x0`.

disable_snooping_dma

Disable DMA snooping.

Type: `bool`. Default value: `0x0`.

force_frame_rate

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.

Type: `int`. Default value: `0x32`.

4.10.72 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 4-493: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PL390_GIC contains the following CADI targets:

- PL390_GIC

PL390_GIC contains the following MTI components:

- [PVBUSlave](#)

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 `pvbust_cpu` and `pvbust_distributor`. The `enable_c<n>` and `match_c<n>` signals identify the originator of a transaction on `pvbust_cpu`. Similarly, `enable_d<n>` and `match_d<n>` signals identify the originator of a transaction on `pvbust_distributor`. `<n>` 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 4-494: Ports

Name	Protocol	Type	Description
<code>cfgsdisable</code>	Signal	Slave	Set preventing write accesses to security-critical configuration registers.
<code>enable_c[8]</code>	ValueState	Slave	Compared with masked PVBUS master id to select processor interface: $(master_id \& enable_c<n>) == match_c<n>$.
<code>enable_d[8]</code>	ValueState	Slave	Compared with masked PVBUS master id to select distributor interface: $(master_id \& enable_d<n>) == match_d<n>$.
<code>legacy_nfiq[8]</code>	Signal	Slave	Legacy FIQ interrupt for processor Interface <code><n></code> .
<code>legacy_nirq[8]</code>	Signal	Slave	Input interrupt signals.
<code>match_c[8]</code>	ValueState	Slave	Mask on the PVBUS master id to select processor interface: $(master_id \& enable_c<n>) == match_c<n>$.
<code>match_d[8]</code>	ValueState	Slave	Mask on the PVBUS master id to select distributor interface: $(master_id \& enable_d<n>) == match_d<n>$.

Name	Protocol	Type	Description
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).
pvbus_cpu	PVBus	Slave	Slave port for connection to processor interface.
pvbus_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

set architecture version in periph_id register.

Type: int. Default value: 0x1.

AXI_IF

set interface type in peripheral identification register 8.

Type: bool. Default value: 0x1.

C_ID_WIDTH

width of the cpu interface master id.

Type: int. Default value: 0x20.

D_ID_WIDTH

width of the distributor interface master id.

Type: int. Default value: 0x20.

ENABLE_LEGACY_FIQ

provide legacy fiq interrupt inputs.

Type: bool. Default value: 0x1.

ENABLE_LEGACY_IRQ

provide legacy irq interrupt inputs.

Type: bool. Default value: 0x1.

ENABLE_PPI_EDGE

ppi edge sensitive.

Type: bool. Default value: 0x0.

ENABLE_TRUSTZONE

support trustzone.

Type: bool. Default value: 0x1.

INIT_ENABLE_C0

initial value of register ENABLE_C0.
Type: int. Default value: 0xffffffff.

INIT_ENABLE_C1

initial value of register ENABLE_C1.
Type: int. Default value: 0xffffffff.

INIT_ENABLE_C2

initial value of register ENABLE_C2.
Type: int. Default value: 0xffffffff.

INIT_ENABLE_C3

initial value of register ENABLE_C3.
Type: int. Default value: 0xffffffff.

INIT_ENABLE_C4

initial value of register ENABLE_C4.
Type: int. Default value: 0xffffffff.

INIT_ENABLE_C5

initial value of register ENABLE_C5.
Type: int. Default value: 0xffffffff.

INIT_ENABLE_C6

initial value of register ENABLE_C6.
Type: int. Default value: 0xffffffff.

INIT_ENABLE_C7

initial value of register ENABLE_C7.
Type: int. Default value: 0xffffffff.

INIT_ENABLE_D0

initial value of register ENABLE_D0.
Type: int. Default value: 0xffffffff.

INIT_ENABLE_D1

initial value of register ENABLE_D1.
Type: int. Default value: 0xffffffff.

INIT_ENABLE_D2

initial value of register ENABLE_D2.
Type: int. Default value: 0xffffffff.

INIT_ENABLE_D3

initial value of register ENABLE_D3.
Type: int. Default value: 0xffffffff.

INIT_ENABLE_D4

initial value of register ENABLE_D4.
Type: int. Default value: 0xffffffff.

INIT_ENABLE_D5

initial value of register ENABLE_D5.
Type: `int`. Default value: `0xffffffff`.

INIT_ENABLE_D6

initial value of register ENABLE_D6.
Type: `int`. Default value: `0xffffffff`.

INIT_ENABLE_D7

initial value of register ENABLE_D7.
Type: `int`. Default value: `0xffffffff`.

INIT_MATCH_C0

initial value of register MATCH_C0.
Type: `int`. Default value: `0x0`.

INIT_MATCH_C1

initial value of register MATCH_C1.
Type: `int`. Default value: `0x1`.

INIT_MATCH_C2

initial value of register MATCH_C2.
Type: `int`. Default value: `0x2`.

INIT_MATCH_C3

initial value of register MATCH_C3.
Type: `int`. Default value: `0x3`.

INIT_MATCH_C4

initial value of register MATCH_C4.
Type: `int`. Default value: `0x4`.

INIT_MATCH_C5

initial value of register MATCH_C5.
Type: `int`. Default value: `0x5`.

INIT_MATCH_C6

initial value of register MATCH_C6.
Type: `int`. Default value: `0x6`.

INIT_MATCH_C7

initial value of register MATCH_C7.
Type: `int`. Default value: `0x7`.

INIT_MATCH_D0

initial value of register MATCH_D0.
Type: `int`. Default value: `0x0`.

INIT_MATCH_D1

initial value of register MATCH_D1.
Type: `int`. Default value: `0x1`.

INIT_MATCH_D2

initial value of register MATCH_D2.
Type: `int`. Default value: `0x2`.

INIT_MATCH_D3

initial value of register MATCH_D3.
Type: `int`. Default value: `0x3`.

INIT_MATCH_D4

initial value of register MATCH_D4.
Type: `int`. Default value: `0x4`.

INIT_MATCH_D5

initial value of register MATCH_D5.
Type: `int`. Default value: `0x5`.

INIT_MATCH_D6

initial value of register MATCH_D6.
Type: `int`. Default value: `0x6`.

INIT_MATCH_D7

initial value of register MATCH_D7.
Type: `int`. Default value: `0x7`.

NUM_CPU

number of cpu interfaces.
Type: `int`. Default value: `0x8`.

NUM_LSPI

number of lockable shared peripheral interrupts.
Type: `int`. Default value: `0x1f`.

NUM_PPI

number of peripheral interrupts.
Type: `int`. Default value: `0x10`.

NUM_PRIORITY_LEVELS

number of priority levels.
Type: `int`. Default value: `0x100`.

NUM_SGI

number of software generated interrupts.
Type: `int`. Default value: `0x10`.

NUM_SPI

number of shared peripheral interrupts.
Type: `int`. Default value: `0x3dc`.

4.10.73 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 4-495: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Ports for PPUMTWakerequest

Table 4-496: Ports

Name	Protocol	Type	Description
cpu_pchannel_m	PChannel	Master	-
ppu_pchannel_s	PChannel	Slave	-
thread_wake_request[2]	Signal	Slave	-
wakerequest	Signal	Master	-

4.10.74 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 4-497: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PPUv0 contains the following CADI targets:

- [PPUv0](#)

PPUv0 contains the following MTI components:

- [PPUv0](#)
- [PVBusSlave](#)

Ports for PPUv0

Table 4-498: 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	-

4.10.75 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 4-499: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

PPUv1 contains the following CADI targets:

- PPUv1

PPUv1 contains the following MTI components:

- [PPUv1](#)
- [PVBusSlave](#)

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 [3.5.1 PChannel protocol](#) on page 95 for further details.

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 `smopen` 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 4-500: Ports

Name	Protocol	Type	Description
dev_clk_en_out	Signal	Master	Domain clock enable
dev_emu_clk_en_out	Signal	Master	Domain emulated mode clock enable
dev_emu_isolaten_out	Signal	Master	Domain emulated isolation control.
dev_isolaten_out	Signal	Master	Domain isolation control.
dev_poresetn_out	Signal	Master	Domain power on reset
dev_ret_resetn_out	Signal	Master	Domain retention reset.
dev_warm_resetn_out	Signal	Master	Domain warm reset
devpactive	PChannel	Master	P-Channel port
irq	Signal	Master	PPU IRQ signal
powerdown	Signal	Master	Notify whether or not the PPU is in OFF state.
ppuhwstat	Value	Master	Notify the power state change inside the PPU. The definition of each bit is the same as DEVPACTIVE[10:0].
pvbuss_s	PVBus	Slave	PPU APB bus slave port
reset_in	Signal	Slave	PPU reset signal input
wakerequest	Signal	Slave	Input port for the wakerequest signal. It is ORed with PACTIVE[8] (ON) inside the PPU as input to PPU DEVPACTIVE[8] (ON). The "is_core_ppu" parameter controls whether there is additional logic to hold this signal until the PPU is in OFF/OFF_EMU state.

Parameters for PPUv1

dbg_recov

Debug Recovery Reset (0: not supported, 1: static mode only, 2: both dynamic & static mode).

Type: int. Default value: 0x0.

default_op_dyn_en

Whether to enable operating mode dynamic transition by default.

Type: bool. Default value: 0x0.

default_op_policy

Default operating policy.

Type: int. Default value: 0x0.

default_power_state_on

Default power state ON.

Type: bool. Default value: 0x0.

default_pwr_dyn_en

Whether to enable dynamic power mode transition by default.

Type: `bool`. Default value: `0x0`.

device_channels

Number of device channels (0: P-Channel, 1-8: Q-Channels).

Type: `int`. Default value: `0x0`.

dynamic_off

Dynamic Off.

Type: `bool`. Default value: `0x0`.

dynamic_on

Dynamic On.

Type: `bool`. Default value: `0x0`.

dynamic_warm_reset

Dynamic Warm Reset.

Type: `bool`. Default value: `0x0`.

full_ret

Full Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

Type: `int`. Default value: `0x0`.

func_ret

Functional Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

Type: `int`. Default value: `0x0`.

is_core_ppu

PPU is core_ppu type which means wake_request would wait till PPU is OFF/OFF_EMU.

Type: `bool`. Default value: `0x0`.

lock_support

Whether to support OFF lock feature.

Type: `bool`. Default value: `0x1`.

logic_ret

Logic Retention (0: not supported, 1: static, 2: dynamic).

Type: `int`. Default value: `0x0`.

mem_off

Memory Off (0: not supported, 1: static mode only, 2: both dynamic & static mode).

Type: `int`. Default value: `0x0`.

mem_ret

Memory Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

Type: `int`. Default value: `0x0`.

mem_ret_emu

Emulated Memory Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

Type: `int`. Default value: `0x0`.

num_opmode_cfg

Number of operating modes.
Type: `int`. Default value: `0x0`.

off_emu

Emulated Off (0: not supported, 1: static mode only, 2: both dynamic & static mode).
Type: `int`. Default value: `0x0`.

op_active_cfg

Operating mode active configuration (0: Ladder use model, 1: Independent user model).
Type: `int`. Default value: `0x0`.

RevD_support

Whether to support Rev D locked IRQ.
Type: `bool`. Default value: `0x1`.

revision

Revision.
Type: `string`. Default value: `"r1p1"`.

use_active_signal

Use device-active signal.
Type: `bool`. Default value: `0x0`.

Related information

[AMBA Low Power Interface Specification Arm Q-Channel and P-Channel Interfaces \(IHI 0068C ID091216\)](#)

4.10.76 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 4-501: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Ports for PPUv1_Cluster_Wakerequest_Logic**Table 4-502: 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	-

Name	Protocol	Type	Description
reset_in	Signal	Slave	-

4.10.77 SMMUv3AEM

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 4-503: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

SMMUv3AEM contains the following CADI targets:

- SMMUv3AEM

SMMUv3AEM contains the following MTI components:

- [PVBusMapper](#)
- [PVBusSlave](#)
- [SMMUv3AEM](#)

Changes in 11.24.4

Parameters added:

- `nw_dcp_extra_drop_conditions`
- `register_accesses_to_root_or_realm_pas_when_no_rme`

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_PMCG_CEID0` 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.

Ports for SMMUv3AEM

Table 4-504: Ports

Name	Protocol	Type	Description
axi_stream_msi_addr_to_match_s	Value_64	Slave	Any SMMU-originated MSI which exactly matches the address in this port will be sent through the AXI stream port axi_stream_msi_m which is usually connected to the GIC through axi_stream_msi_s. As MSIs are 32 bit aligned then if bits[1:0] != 0 or the address is above OAS then this is effectively disabled. The parameter smmu_msi_device_id is the DeviceID to send on the interface. See also the parameters: axi_stream_msi_TID and axi_stream_msi_TDEST. The default value of this port is set by the parameter axi_stream_msi_addr_to_match. This port is sampled at negedge of reset_in.
axi_stream_msi_m	PVBus	Master	Manager port used for sending SMMU originated MSIs directly to the GIC when axi_stream_msi_enabled == true
clk_in	ClockSignal	Slave	Clock signal
conf_reset_of_SMMU_GBPA_ABORT	Signal	Slave	System reset value of SMMU_GBPA.ABORT.
conf_reset_of_SMMU_S_GBPA_ABORT	Signal	Slave	System reset value of SMMU_S_GBPA.ABORT.
conf_system_supports_bgptm	Signal	Slave	System supports broadcast TLBI PAALL and TLBI RPA for supporting RME.
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.
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.
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.
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.
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.

Name	Protocol	Type	Description
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.
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.

Name	Protocol	Type	Description
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.
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 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 pin and legacy_tz_en is low then the model will obey the setting of the parameter out_of_range_logptsz_s. This signal is sampled at negedge of reset_in.
legacy_tz_en	Signal	Slave	For an RME-enabled SMMU then tie this high to get non-RME behaviour See also the parameter SMMU_ROOT_IDRO. This is reset-edge sampled and so if you drive it then you must drive it _before_ the negedge of the reset.
pvbust_control_s	PVBus	Slave	Register subordinate port

Name	Protocol	Type	Description
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. 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 pvbus_s[] port.
pvbus_m_tw_msi_qs	PVBus	Master	Manager port used for Table Walks, MSIs and Queue access when separate_tw_msi_qs_port==true
pvbus_s[64]	PVBus	Slave	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered pvbus_m[] port.
reset_in	Signal	Slave	Reset signal
sev_out	Signal	Master	Event signal

Parameters for SMMUv3AEM

all_error_messages_through_trace

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. .

Type: bool. Default value: 0x0.

allow_non_secure_access_to_SMMU_S_INIT

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. .

Type: bool. Default value: 0x0.

apply_ste_instcfg_privcfg_on_all_ats_translated_accesses

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. .

Type: bool. Default value: 0x0.

ats_split_stage_dbm_update_do_with_ATSRequest

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. .

Type: int. Default value: 0x0.

axi_stream_msi_addr_to_match

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 that 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`.

Type: `int`. Default value: `0x1`.

axi_stream_msi_TDEST

ID of the AXI stream subordinate port on the GIC that receives SMMU originated MSIs sent directly. The name of the GIC port is `axi_stream_msi_s`. NOTE that 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`.

Type: `int`. Default value: `0x0`.

axi_stream_msi_TID

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 that 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`.

Type: `int`. Default value: `0x0`.

behaviour_of_sampled_at_reset_signals

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. .

Type: `int`. Default value: `0x0`.

cmdq_max_number_of_commands_to_buffer

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 will accentuate this effect. .

Type: `int`. Default value: `0xa`.

dpt_configure_ATS_formed_entries

For ATS-formed DPT TLB entries then a comma-separated list of options. Some options are mutually exclusive. If the string is empty or "none", then no ATS-formed entries are inserted into the DPT. Otherwise... One of: `* "all_enabled_stages" / ""` -- use all enabled VMSA stages (default) `* "last_enabled_stage"` -- use only the last enabled VMSA stage. The region stored in the DPT entry might be just the VMSA region or the VMSA region truncated by the GPC checked region. One of: `* "vmsa_and_gpc" / ""` -- store as the smaller of the VMSA and GPC region (default) `* "vmsa_only"` -- store as the VMSA region. The implementation may distinguish ATS-formed entries and DPT-formed entries: `* "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. Example: "all_enabled_stages, vmsa_and_gpc".

Type: string. Default value: "all_enabled_stages, vmsa_and_gpc".

dpt_configure_invalidation

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 .

Type: string. Default value: "".

enable_device_id_checks

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 then if it overflows 32 bits then the model will warn. If the DeviceID is not used then it is assumed that the external agent that forms the DeviceID will warn if it overflows. * If the SMMU supports MSIs, then the model will check that the GIC will be 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. .

Type: bool. Default value: 0x1.

hide_warning_EOPD_differs_from_what_would_be_cached

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. .

Type: bool. Default value: 0x0.

hide_warning_NoStreamID_transaction_for_unsupported_PAS_or_MPAM_SP

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 will warn when this occurs, but the warning can be hidden by setting this parameter. .

Type: bool. Default value: 0x0.

howto_identify

If 'use-identify' then will use 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. The StreamID is secure if SEC_SID is true. The SubstreamID (20 b) is valid if SSV is true. NoStreamID transactions are identified by SIDV == 0 and the SSD is the PAS of the transaction and SEC_SID is not used. nSEC_SID, nSSV, nSIDV are available with negative logic. Negative and positive logic symbols for the same

attribute is a error. Different attributes are independent and can use negative or positive logic. .

Type: `string`. Default value: `"use-identify"`.

howto_identify_NoStreamID_extra_info

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] .

Type: `string`. Default value: `""`.

httu_early_st2_permission_fault_if_af_update_at_stage1

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. .

Type: `int`. Default value: `0x0`.

httu_memory_types_supported

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 .

Type: `string`. Default value: `"rawaWB, raWB, waWB, naWB"`.

imp_def_alloccfg

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 .

Type: `int`. Default value: `0x0`.

imp_def_ats_attribute_stashing

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. .
Type: `int`. Default value: `0x0`.

`imp_def_cohacc_effect`

SMMU_IDR0.COHAAC is a system property. However, the exact nature of the transactions that the SMMU emits is an IMP DEF property when COHAAC == 0: 0 -- COHAAC == 0 forces the output attributes of SMMU-generated accesses to non-shared. 1 -- The only effect of COHAAC is what is reported in SMMU_IDR0.COHAAC and has no effect on the output attributes of SMMU-generated accesses. .
Type: `int`. Default value: `0x0`.

`imp_def_contiguous_bit_handling`

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 .
Type: `int`. Default value: `0x0`.

`imp_def_effective_ATTR_TYPES_OVR_is_false_per_port`

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 .
Type: `string`. Default value: `""`.

`imp_def_has_PID_CID`

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. .
Type: `bool`. Default value: `0x1`.

`imp_def_L1CD_L2Ptr_out_of_range`

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. .
Type: `int`. Default value: `0x0`.

imp_def_ns_bit_for_s_gatos_on_s1_bypass_non_sel2

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. .

Type: int. Default value: 0x0.

imp_def_ns_bit_for_s_vatos_on_s1_bypass

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. .

Type: int. Default value: 0x0.

imp_def_PID0

If imp_def_has_PID_CID is true then this is the PID0 value. .

Type: int. Default value: 0x83.

imp_def_PID1

If imp_def_has_PID_CID is true then this is the PID1 value. .

Type: int. Default value: 0xb4.

imp_def_PID2

If imp_def_has_PID_CID is true then this is the PID2 value. .

Type: int. Default value: 0xb.

imp_def_PID3

If imp_def_has_PID_CID is true then this is the PID3 value. .

Type: int. Default value: 0x0.

imp_def_PID4

If imp_def_has_PID_CID is true then this is the PID4 value. .

Type: int. Default value: 0x4.

imp_def_ras_access_control_policy

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" .

Type: string. Default value: "use-

imp_def_ras_allow_non_secure_accesses_if_supports_secure".

imp_def_ras_allow_non_secure_accesses_if_supports_secure

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. .

Type: bool. Default value: 0x0.

imp_def_reset_unknown_fields_to_zero

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. .

Type: bool. Default value: 0x0.

imp_def_rme_gpf_syndrome_for_PMCG_MSIs

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 .

Type: string. Default value: "other_gpf".

imp_def_rme_gpf_syndrome_for_RAS_MSIs

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 .

Type: string. Default value: "other_gpf".

imp_def_rme_mpam_info_from_NoStreamID_on_gpt_walks_ignored

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 .

Type: bool. Default value: 0x0.

imp_def_S1ContextPtr_out_of_range

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. .

Type: int. Default value: 0x0.

imp_def_split_ATS_attributes_is_stage1

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. .

Type: `bool`. Default value: `0x0`.

imp_def_truncate_out_of_range_streamids_on_invalidate_commands

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. .

Type: `bool`. Default value: `0x0`.

imp_def_v3_atos_fault

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) .

Type: `int`. Default value: `0x0`.

ish_is_osh

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. .

Type: `bool`. Default value: `0x0`.

mec_attribute_transform

If MEC is supported, this is applied to `_all_` downstream transactions to transport the MEC information. * `""/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. NOTE: * attribute transforms applied before this: * for client transactions `'output_attribute_transform'/'output_attribute_transform_for_NoStreamID'`. * for table walks `'tw_qs_attribute_transform'`. * for MSIs `'msi_attribute_transform'`. * if MPAM is enabled `'mpam_attribute_transform'`. .

Type: `string`. Default value: `""`.

mpam_attribute_transform

If MPAM is supported, this is applied to `_all_` downstream transactions to transport the MPAM information. * `""/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`. .

Type: `string`. Default value: `"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS"`.

msi_attribute_transform

Transform downstream attributes of MSI transactions. * ""/"none" -- no transform * How to alter output attributes of SMMU-generated MSIs. Example: "UserFlags[15:0]=smmu_msi_device_id[31:16], MasterID[15:0]=smmu_msi_device_id[15:0], ExtendedID=0" RHS/LHS Symbols: * ExtendedID/MasterID/UserFlags RHS Symbols: * the parameter smmu_msi_device_id * the symbol 'interrupt_kind' * 0/1 -- EVENTQ s/ns * 2 -- PRIQ * 3/4 -- CMD_SYNC s/ns * 5/6 -- GERROR s/ns * 7/8 -- PMCG s/ns * 9/10/11 -- RAS FHI/ERI/CRI * HWATTR_KIND_0: PBHA information * numeric literals. ExtendedID/MasterID/UserFlags start with values {0, 0xffffffff, 0} respectively. Any bits with no transform are unchanged. This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices. NOTE: * see also 'output_attribute_transform' and enable_device_id_checks. . Type: string. Default value: "ExtendedID[31:0]=smmu_msi_device_id, MasterID=0xffffffff".

msi_ra_wa_tr

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. . Type: int. Default value: 0x7.

non_arch_incoming_stronger_than_iWB_oWB_forces_output_iNC_oNC_or_stronger

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 . Type: string. Default value: "".

normalize_input_normal_non_iWB_oWB_to_iNC_oNC_osh

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. . Type: bool. Default value: 0x0.

number_of_ports

The number of port pairs that the SMMU has. . Type: int. Default value: 0x1.

nw_dcp_extra_drop_conditions

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. .
Type: `string`. Default value: "".

ordering_of_PAN_and_xn_by_ns_pas

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. .
Type: `int`. Default value: 0x0.

out_of_range_CMD_ATC_INV_Size

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. .
Type: `int`. Default value: 0x0.

out_of_range_logptsz_s

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. .
Type: `int`. Default value: -0x1.

output_attribute_transform

Transform downstream attributes of StreamID transactions. * ""/"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. .

Type: string. Default value: "ExtendedID[31:0]=DeviceID".

output_attribute_transform_for_NoStreamID

Transform downstream attributes of NoStreamID transactions. * ""/"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. .

Type: string. Default value: "ExtendedID[31:0]=0, ExtendedID[32]=1".

output_id_routed_transform

The SMMU generates the following ID-routed transaction on the pvtbus_id_routed_m bus:
* ATC Invalidate * PRI Response This parameter expresses 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 .

Type: string. Default value: "Address[43:12]=StreamID, PAS=SSD".

percent_commit

Percentage of times that a read of a register with Update will commit the update. 0 means commit immediately. .

Type: int. Default value: 0x14.

percent_commit_Update_clear

Percentage of times that a read of a register with a pending Update clear will lower the Update flag. .

Type: int. Default value: 0x14.

pmu

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. .

Type: string. Default value: "".

ports_that_ignore_PnU_InD_on_transactions_with_no_SubstreamID

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. .

Type: `string`. Default value: "".

prefetch_only_requests

The simulator supports 'prefetch-only' DMI requests, which can occur for anytime 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()`. .

Type: `int`. Default value: 0x0.

PRESET_REL_base_address

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) .

Type: `int`. Default value: 0x0.

QUEUES_PRESET_smmu_cmdq_base

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. .

Type: `int`. Default value: 0x0.

QUEUES_PRESET_smmu_eventq_base

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. .

Type: `int`. Default value: 0x0.

QUEUES_PRESET_smmu_priq_base

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. .

Type: `int`. Default value: 0x0.

QUEUES_PRESET_smmu_r_cmdq_base

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. .

Type: `int`. Default value: `0x0`.

QUEUES_PRESET_smmu_r_eventq_base

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. .

Type: `int`. Default value: `0x0`.

QUEUES_PRESET_smmu_r_priq_base

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. .

Type: `int`. Default value: `0x0`.

QUEUES_PRESET_smmu_s_cmdq_base

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. .

Type: `int`. Default value: `0x0`.

QUEUES_PRESET_smmu_s_eventq_base

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. .

Type: `int`. Default value: `0x0`.

ras

What to instantiate for RAS handling. Values of this parameter are: * "" -- no RAS records
* "mmu_600" * "mmu_700" mmu_600: * only a corrected errors reported. See also `imp_def_ras_allow_non_secure_accesses_if_supports_secure`. .

Type: `string`. Default value: "".

register_accesses_to_root_or_realm_pas_when_no_rme

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 .

Type: `int`. Default value: `0x0`.

reset_value_of_SMMU_GBPA

Reset value of SMMU_GBPA .

Type: `int`. Default value: `0x0`.

reset_value_of_SMMU_S_GBPA

Reset value of SMMU_S_GBPA .

Type: `int`. Default value: `0x0`.

rme_ats_request_pa_strategy

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. .

Type: `int`. Default value: 0x0.

rme_da_force_better_configuration

RME-DA requires that the SMMU be integrated into a system for which SMMU_IDR0.COHAACC == 1 and SMMU_IDR0.IDR0.HTTU == both_af_and_dirty (2). The model has pins: * `conf_system_supports_cohacc` * `conf_system_supports_httu` that can control these ID fields. In addition, RME-DA requires that the fundamental SMMU has certain properties that are configured by its ID codes. This parameter allows you to selectively ignore the pins and bad ID to produce a good configuration by forcing the required values. Whether the SMMU has RME-DA or not is identified by SMMU_ROOT_IDR0.REALM_IMPL. This is a comma-separated list of fields to force when RME-DA is configured by SMMU_ROOT_IDR0.REALM_IMPL: * in SMMU_IDR0: * "Hyp" * "S1P" * "S2P" * "TTF" * "NS1ATS" * "COHAACC" * "HTTU" * "RME_IMPL" * "SSIDSIZE" in SMMU_IDR1 * "BBML" in SMMU_IDR3 You can also use "all" to set all. .

Type: `string`. Default value: "".

rme_l0gpt_entry_covers_log2size_in_bytes

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. .

Type: `int`. Default value: 0x1e.

rme_speculation_control

This is a comma-separated list of flags that control when and how the model will perform speculation for RME. .

Type: `string`. Default value: "".

root_register_page_offset

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. .

Type: `int`. Default value: 0x0.

secure_state_controls_access_to_SMMU_S_INIT

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. .

Type: `bool`. Default value: 0x1.

seed

Used to seed the pseudo-random number generator that the SMMU model uses. .

Type: `int`. Default value: 0x12345678.

separate_tw_msi_qs_port

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. .

Type: `bool`. Default value: `0x1`.

size_of_cd_cache

The number of entries in the cache holding CD structures. 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. .

Type: `int`. Default value: `0x0`.

size_of_dpttlb

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. .

Type: `int`. Default value: `0x0`.

size_of_gpttlb

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. .

Type: `int`. Default value: `0x0`.

size_of_llcd_cache

The number of entries in the cache holding L1CD descriptors. 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. .

Type: `int`. Default value: `0x0`.

size_of_llste_cache

The number of entries in the cache holding L1STE descriptors. 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. .

Type: `int`. Default value: `0x0`.

size_of_register_file

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. .

Type: `int`. Default value: `0x100000`.

size_of_ste_cache

The number of entries in the cache holding STE structures. 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. .

Type: `int`. Default value: `0x0`.

size_of_tlb

The number of entries in the 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. .

Type: `int`. Default value: `0x0`.

SMMU_AIDR

SMMU_AIDR contains the Major and Minor architectural revisions numbers .

Type: `int`. Default value: `0x0`.

SMMU_IDR0

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`. .

Type: `int`. Default value: `0x0`.

SMMU_IDR1

SMMU_IDR1. .

Type: `int`. Default value: `0x0`.

SMMU_IDR2

SMMU_IDR2 holds the BA_VATOS field. .

Type: `int`. Default value: `0x0`.

SMMU_IDR3

SMMU_IDR3 is reserved. .

Type: `int`. Default value: `0x0`.

SMMU_IDR4

SMMU_IDR4 is Imp def. .

Type: `int`. Default value: `0x0`.

SMMU_IDR5

SMMU_IDR5 contains, amongst others the output address encoded size (OAS). .

Type: `int`. Default value: `0x0`.

SMMU_IDR6

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. .

Type: `int`. Default value: `0x0`.

SMMU_IIDR

SMMU_IIDR contains fields for the implementer, product revision, etc. .

Type: `int`. Default value: `0x0`.

SMMU_MPAMIDR

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. .

Type: `int`. Default value: `0x0`.

smmu_msi_device_id

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID. See parameter `msi_attribute_transform` and `enable_device_id_checks`. .

Type: `int`. Default value: `0x0`.

SMMU_R_AIDR

The value of SMMU_R_AIDR.

Type: `int`. Default value: `0x0`.

SMMU_R_IDR0

The value of SMMU_R_IDR0. .

Type: `int`. Default value: `0x0`.

SMMU_R_IDR3

The value of SMMU_R_IDR3. .

Type: `int`. Default value: `0x0`.

SMMU_R_IDR6

The value of SMMU_R_IDR6 that configures the ECMDQs. .

Type: `int`. Default value: `0x0`.

SMMU_R_MECIDR

The value of SMMU_R_MECIDR. .

Type: `int`. Default value: `0x0`.

SMMU_R_MPAMIDR

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. .

Type: `int`. Default value: `~0x1`.

SMMU_ROOT_IDR0

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`. .

Type: `int`. Default value: `0x0`.

SMMU_ROOT_IIDR

The value of the SMMU_ROOT_IIDR register. If is zero then will be the same as SMMU_IIDR. .

Type: `int`. Default value: `0x0`.

SMMU_S_IDR0

Secure IDR0 register. .

Type: `int`. Default value: `0x0`.

SMMU_S_IDR1

SMMU_S_IDR1 Indicates if there is a secure side by bit 31. .

Type: `int`. Default value: `0x0`.

SMMU_S_IDR2

SMMU_S_IDR2 Reserved .

Type: `int`. Default value: `0x0`.

SMMU_S_IDR3

SMMU_S_IDR3 Reserved .

Type: `int`. Default value: `0x0`.

SMMU_S_IDR4

SMMU_S_IDR4 IMP DEF .

Type: `int`. Default value: `0x0`.

SMMU_S_IDR6

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. .

Type: `int`. Default value: `0x0`.

SMMU_S_MPAMIDR

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. .

Type: `int`. Default value: `0x0`.

smmu_v33_begin_offset_of_qcp0

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. .

Type: `int`. Default value: `0x0`.

support_for_httu_when_starts_disallowed

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 .

Type: `int`. Default value: `0x0`.

TABLES_PRESET_smmu_r_strtab_base

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. .

Type: `int`. Default value: `0x0`.

TABLES_PRESET_smmu_r_strtab_base_cfg

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. .

Type: int. Default value: 0x0.

TABLES_PRESET_smmu_s_strtab_base

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. .

Type: int. Default value: 0x0.

TABLES_PRESET_smmu_s_strtab_base_cfg

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. .

Type: int. Default value: 0x0.

TABLES_PRESET_smmu_strtab_base

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. .

Type: int. Default value: 0x0.

TABLES_PRESET_smmu_strtab_base_cfg

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. .

Type: int. Default value: 0x0.

tlb_when_do_f_tlb_conflict_on_overlap

If a TLB entry is created by a walk and it overlaps an existing entry. Then 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. .

Type: int. Default value: 0x0.

translated_device_id_base

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. .

Type: int. Default value: 0x0.

treat_debug_read_accesses_as_speculative_accesses

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. .

Type: bool. Default value: 0x0.

tw_qs_attribute_transform

Transform downstream attributes of table walk and queue transactions. *

""/"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 * numeric literals.

ExtendedID/MasterID/UserFlags start with values {0, 0xFFFFFFFF, 0} respectively. Any bits with no transform are unchanged. NOTE: * see also 'output_attribute_transform' and 'msi_attribute_transform'.

Type: string. Default value: "".

unpred_httu_percent_do_discretionary_AF

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. .

Type: int. Default value: 0x32.

unpred_httu_percent_do_discretionary_DBM

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. .

Type: int. Default value: 0x32.

unpred_translated_access_out_of_range_of_oas

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 .

Type: int. Default value: 0x1.

wait_atos_ticks

This is the time to wait before doing an ATOS operation. If bit 32 is set (0x1_0000_0000)

then the time waited for is a uniform randomly distributed time [0,max(2,(t & 0xFFFFFFFF))). .

Type: int. Default value: 0x0.

wait_cmdq_ticks

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))). .

Type: int. Default value: 0x0.

wait_eventq_ticks

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))). .

Type: int. Default value: 0x0.

wait_misc_async_actions_ticks

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))). .

Type: int. Default value: 0x0.

wait_msi_ticks

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))). .

Type: `int`. Default value: 0x0.

wait_pri_req_ticks

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))). .

Type: `int`. Default value: 0x0.

wait_pri_resp_ticks

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, that immediately makes a PRI Request, that 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))). .

Type: `int`. Default value: 0x1.

when_fetch_vms

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. .

Type: `int`. Default value: 0x0.

width_of_agbpa_impdef

Width of the SMMU_s_AGBPA.IMPDEF field. .

Type: `int`. Default value: 0x10.

4.10.78 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 4-505: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

SMSC_91C111 contains the following CADI targets:

- SMSC_91C111

SMSC_91C111 contains the following MTI components:

- [PVBusSlave](#)

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.

Ports for SMSC_91C111

Table 4-506: 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.
pvbuss	PVBus	Slave	Slave port for register access.
state	ValueState_64	Master	State port to retrieve state of host bridge

Related information

[Configuring the networking environment for Microsoft Windows](#) on page 53

[Configuring the networking environment for Linux](#) on page 59

4.10.79 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 4-507: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

SP804_Timer contains the following CADI targets:

- ClockDivider

- CounterModule
- SP804_Timer

SP804_Timer contains the following MTI components:

- [ClockDivider](#)
- [PVBusSlave](#)

Ports for SP804_Timer

Table 4-508: 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.

4.10.80 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 4-509: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

SP805_Watchdog contains the following CADI targets:

- ClockTimerThread
- ClockTimerThread64
- SP805_Watchdog
- SchedulerThread
- SchedulerThreadEvent

SP805_Watchdog contains the following MTI components:

- [PVBusSlave](#)

Ports for SP805_Watchdog

Table 4-510: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock input, typically 1MHz, driving master count rate.
irq_out	Signal	Master	Interrupt signaling.
pvbuss	PVBus	Slave	Slave port for register access.
reset_in	Signal	Slave	Reset signaling.
reset_out	Signal	Master	Reset signaling.

Parameters for SP805_Watchdog

simhalt

Halt on reset.

Type: bool. Default value: 0x0.

4.10.81 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 4-511: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

SP810_SysCtrl contains the following CADI targets:

- ClockDivider
- SP810_SysCtrl

SP810_SysCtrl contains the following MTI components:

- [ClockDivider](#)
- [PVBusSlave](#)

Ports for SP810_SysCtrl

Table 4-512: 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.

Name	Protocol	Type	Description
pll_en	Signal	Master	PLL enable output. Not fully implemented. Using this port has unpredictable results.
pvbus	PVBus	Slave	Slave port for register access.
ref_clk_in	ClockSignal	Slave	Clock source used by the Timer and Watchdog modules.
remap_clear	StateSignal	Master	Remap clear request output.
remap_stat	StateSignal	Slave	Remap status input. Not fully implemented. Using this port has unpredictable results.
sleep_mode	Signal	Master	Control clocks for SLEEP mode. Not fully implemented. Using this port has unpredictable results.
sys_id	ValueState	Slave	Unused port.
sys_mode	ValueState	Slave	Present system mode. Not fully implemented. Using this port has unpredictable results.
sys_stat	ValueState	Slave	System status input. Not fully implemented. Using this port has unpredictable results.
timer_clk_en[4]	ClockRateControl	Master	Timer clock enable n.
wd_clk_en	Signal	Master	Watchdog module clock enable output. Not fully implemented. Using this port has unpredictable results.
wd_en	Signal	Slave	Watchdog module enable input. Not fully implemented. Using this port has unpredictable results.

Parameters for SP810_SysCtrl

sysid

System Identification Register.
Type: int. Default value: 0x0.

use_s8

Use Switch 8 (S1-S4).
Type: bool. Default value: 0x0.

4.10.82 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 4-513: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

TZC_400 contains the following CADI targets:

- TZC_400

TZC_400 contains the following MTI components:

- [PVBusMapper](#)
- [PVBusSlave](#)

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.



Note

- 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.

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.

Ports for TZC_400

Table 4-514: 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.

4.10.83 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 4-515: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

TZFilterUnit contains the following CADI targets:

- TZFilterUnit

TZFilterUnit contains the following MTI components:

- [PVBusMapper](#)

Ports for TZFilterUnit

Table 4-516: Ports

Name	Protocol	Type	Description
control	TZFilterControl	Master	Configuration port.
pvbus_m	PVBus	Master	Master bus port.
pvbus_s	PVBus	Slave	Slave bus port.

4.10.84 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 4-517: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

TZIC contains the following CADI targets:

- TZIC

TZIC contains the following MTI components:

- [PVBusSlave](#)

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 4-518: 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.
pvbuse	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.

4.10.85 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 4-519: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Ports for VHT_VIOBridge

Table 4-520: Ports

Name	Protocol	Type	Description
pvbuss_s	PVBus	Slave	Target port for access to VIOBridge registers

4.10.86 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 4-521: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Ports for VHT_VSIBridge

Table 4-522: Ports

Name	Protocol	Type	Description
intr	Signal	Master	Interrupt raising signal
pvbuss_m	PVBus	Master	Requester port for access to emeory
pvbuss_s	PVBus	Slave	Completer port for access to VSIBridge registers

4.10.87 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 4-523: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

Ports for VHT_VSocket

Table 4-524: Ports

Name	Protocol	Type	Description
pvbush_m	PVBus	Master	Port for VSocket bridge to access external memory
pvbush_s	PVBus	Slave	Target port for access to VSocket bridge registers

4.10.88 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 4-525: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [2.3 Quality level definitions](#) on page 28.

v7_VGIC contains the following CADI targets:

- v7_VGIC

v7_VGIC contains the following MTI components:

- [PVBusSlave](#)
- [v7_VGIC](#)

Ports for v7_VGIC

Table 4-526: 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.

Name	Protocol	Type	Description
ppi_core1[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_core2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_core3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_core4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_core5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_core6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_core7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.
pvbus_s	PVBus	Slave	Bus port for accessing distributor registers.
reporting_interface	VGICReportingProtocol	Slave	Logging interface.
reset_signal	Signal	Slave	Reset signal input.
spi[988]	Signal	Slave	SPI inputs.
vfiq_out[8]	Signal	Master	Virtual FIQ outputs.
virq_out[8]	Signal	Master	Virtual IRQ outputs.
wakeup_fiq[8]	Signal	Master	Wakeup signal for FIQ.
wakeup_irq[8]	Signal	Master	Wakeup signal for IRQ.

Parameters for v7_VGIC

enable_log_errors

Enable logging of errors.

Type: bool. Default value: 0x0.

enable_log_fatal

Enable logging of fatal errors.

Type: bool. Default value: 0x0.

enable_log_warnings

Enable logging of warnings.

Type: bool. Default value: 0x0.

enabled

Enable the component. If it is disabled then all register writes will have no effect.

Type: bool. Default value: 0x1.

5. 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/`.

5.1 Loading a plug-in

The method of loading a plug-in depends on the type of model being used.

5.1.1 --plugin command-line option

For an FVP, or an EVS that uses the `scx::scx_parse_and_configure()` method, or when using Model Shell or Model Debugger, specify the plug-in using the `--plugin <path_to_plugin>/<plugin_name>` switch.

If more than one plug-in is required, use multiple `--plugin` switches.

Specify plug-in parameters using the `-c` option when launching the platform model, or in a configuration file.

For example, to load the `TarmacTrace` plug-in on Windows:

1. The following command lists both the model parameters and the plug-in parameters:

```
FVP_Base_AEMvA.exe --plugin="%PVLIB_HOME%\plugins\Win64_VC2019\Release\TarmacTrace.dll" --list-params
```

2. The following command:

- Loads the `TarmacTrace` plug-in.
- Specifies the name of the log file using the plug-in parameter `TRACE.TarmacTrace.trace-file`.

```
FVP_Base_AEMvA.exe ... \  
--plugin="%PVLIB_HOME%\plugins\Win64_VC2019\Release\TarmacTrace.dll" \  
-C TRACE.TarmacTrace.trace-file=trace.log \  
...
```

Related information

[scx::scx_parse_and_configure](#)

5.1.2 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/<OS_Compiler>/<plug-in_name>.so");
```

Related information

[scx::scx_load_plugin](#)

5.1.3 FM_TRACE_PLUGINS environment variable

When 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 a semicolon, 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 the plug-in instance a name by adding the prefix `<instancename>=` to the 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/myname/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
```



Note

- Do not specify a pipe character `|` at the end of the environment variable value.
- Separate individual parameters with two pipe characters `||`, not one.
- Separate the plug-in from the parameters using a single pipe character `|`.
- Do not specify quotes around paths that contain spaces, or escape spaces. Spaces are resolved automatically.
- Do not specify environment variables within this environment variable.
- Specify the parameters before you specify the plug-in.
- On Windows, you can either set the environment variable using the **Advanced System Settings** window, or if you are using the command line, you might need to escape each pipe character, for example `^^|`.
- The parameter values that you set using this environment variable are not displayed by the `--list-params` or `-l` command-line options. This is because the environment variable is processed at a later stage, when launching the simulation.

5.2 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>
```

For trace plug-ins, the plug-in type prefix is `TRACE`, for example:

```
-C TRACE.TarmacTrace.trace-file=trace.log
```

The prefix varies for other plug-in types, for example:

```
-C SVE.ScalableVectorExtension.clear_constrained_lanes=1
```

5.3 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. They 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, triggering 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 1651

5.3.1 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 5-1: ArchMsgTrace parameters

Name	Type	Default value	Allowed values	When set	Description
<code>exit_on_first_output</code>	bool	false	true, false	Runtime	Exit the simulation process after the first message has been written.
<code>filter_tags</code>	string	"ALL"	""	Runtime	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.
<code>suppress_categories</code>	string	"why"	""	Runtime	Space-separated list of categories which should not be printed.

Name	Type	Default value	Allowed values	When set	Description
suppress_repeated	bool	true	true, false	Runtime	Suppress repeated messages from similar call sites.
suppress_sources	string	""	""	Runtime	Space-separated list of components or events that should not be printed.
trace-file	string	""	""	Runtime	ArchMsgTrace output file.

5.4 ASTFplugin

ASTFplugin is an MTI plug-in that adds support to Fast Models for generating 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.



Note

- ASTFplugin and the ASTF specification are in development and further iterations are expected. For the status of ASTFplugin in this release, the version of the specification it supports, and any limitations and known issues, see the Fast Models Portfolio release notes.
- ASTFplugin can be used with [5.20 ToggleMTIPlugin](#) on page 1685.

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](#).

5.4.1 ASTFplugin usage notes

Be aware of the following 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>
```

For the list of ASTFplugin parameters, see [5.4.4 ASTFplugin - parameters](#) on page 1624.

- 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.
- If parameter `TRACE.ASTFplugin.timestamp-enable` is set to 1, timing annotation must be enabled, or the plug-in outputs an error message and terminates the simulation. Timing annotation is enabled by default. It is disabled when environment variable `FASTSIM_DISABLE_TA` is set to 1.
- ASTFplugin tries to register callbacks for MTI trace sources for the Scalable Vector Extension (SVE). If `scalableVectorExtension.so` 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 enable SVE operations, load `scalableVectorExtension.so` using the `--plugin` option. To record SVE operations, `scalableVectorExtension.so` must be loaded before `ASTFplugin.so`.
- 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.

Related information

[Timing Annotation](#)

5.4.2 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 [5.4.3 ASTF tools](#) on page 1624.
- [5.20 ToggleMTIPlugin](#) on page 1685, 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 interface 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](#)

5.4.3 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`.

5.4.4 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 5-2: 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	Runtime	Timestamp records will become part of the output if enabled.
timestamp-period	int	0x5	0x1 - 0x7fffffffffffffffff	Runtime	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.

5.5 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.



Note

CortexA53Predictor

Cortex®-A53 branch predictor. This is the default.

5.5.1 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 5-3: BranchPrediction parameters

Name	Type	Default value	Allowed values	Runtime	Description
bpstat-branchcount	int	-0x1	0x0 - 0x7fffffffffffffffff	false	The number of branch instructions to display. Set to -1 to display all branch instructions.
bpstat-pathfilename	string	-	-	false	The path and filename of the branch statistics log, relative to the current directory. If you omit it, no log file is generated.
mispredict-latency	int	0x8	0x0 - 0x7fffffffffffffffff	false	The number of instructions that are flushed for every misprediction. This value is equal to the pipeline length of the processor. For example, in a processor where the prefetch unit has three stages and the execution units have five stages, the pipeline length is eight. Therefore the <code>mispredict-latency</code> is eight clock cycles.
predictor-type	string	CortexA53Predictor	See the note after this table.	false	The type of branch predictor to use.

5.5.2 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

```

5.5.3 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
```

5.6 CADIIPCRemoteConnection

Use this plug-in to enable a remote CADI connection to be made to a model.

The remote connection can be from a client on a different host machine, or 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, a CADI server must be running, specified 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](#)

5.6.1 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 5-4: CADIIPCRemoteConnection parameters

Name	Type	Default value	Allowed values	When set	Description
<code>enable_remote_cadi</code>	bool	false	true, false	Runtime	Allow connections from remote hosts.
<code>listen_address</code>	string	"127.0.0.1"	""	Runtime	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	Runtime	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	Runtime	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.

5.7 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:

- [4.5.49 ARMCortexM33CT](#) on page 815
- [4.5.51 ARMCortexM52CT](#) on page 835
- [4.5.52 ARMCortexM55CT](#) on page 837
- [4.5.53 ARMCortexM85CT](#) on page 850
- [4.5.3 ARMAEMv8MCT](#) on page 252

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.

5.7.1 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
```

5.7.2 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 5-5: CDETester parameters

Name	Type	Default value	Allowed values	When set	Description
<code>cde_tester_trivial.cps_implemented</code>	int	0xff00ff	0x0 - 0xffffffff	Runtime	Bitmask indicating coprocessors implemented.
<code>cde_tester_trivial.cps_implemented_cx1</code>	int	0xff00ff	0x0 - 0xffffffff	Runtime	Coprocessor enables CX1 instructions ([23:16] for CX1A, [7:0] for CX1).
<code>cde_tester_trivial.cps_implemented_cx1d</code>	int	0xff00ff	0x0 - 0xffffffff	Runtime	Coprocessor enables CX1D instructions ([23:16] for CX1DA, [7:0] for CX1D).
<code>cde_tester_trivial.cps_implemented_cx2</code>	int	0xff00ff	0x0 - 0xffffffff	Runtime	Coprocessor enables CX2 instructions ([23:16] for CX2A, [7:0] for CX2).
<code>cde_tester_trivial.cps_implemented_cx2d</code>	int	0xff00ff	0x0 - 0xffffffff	Runtime	Coprocessor enables CX2D instructions ([23:16] for CX2DA, [7:0] for CX2D).
<code>cde_tester_trivial.cps_implemented_cx3</code>	int	0xff00ff	0x0 - 0xffffffff	Runtime	Coprocessor enables CX3 instructions ([23:16] for CX3A, [7:0] for CX3).

Name	Type	Default value	Allowed values	When set	Description
cde_tester_trivial.cps_implemented_cx3d	int	0xff00ff	0x0 - 0xffffffff	Runtime	Coprocessor enables CX3D instructions ([23:16] for CX3DA, [7:0] for CX3D).
cde_tester_trivial.cps_implemented_vcx1	int	0xff00ff	0x0 - 0xffffffff	Runtime	Coprocessor enables VCX1 instructions ([23:16] for VCX1A, [7:0] for VCX1).
cde_tester_trivial.cps_implemented_vcx1v	int	0xff00ff	0x0 - 0xffffffff	Runtime	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	Runtime	Coprocessor enables VCX2 instructions ([23:16] for VCX2A, [7:0] for VCX2).
cde_tester_trivial.cps_implemented_vcx2v	int	0xff00ff	0x0 - 0xffffffff	Runtime	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	Runtime	Coprocessor enables VCX3 instructions ([23:16] for VCX3A, [7:0] for VCX3).
cde_tester_trivial.cps_implemented_vcx3v	int	0xff00ff	0x0 - 0xffffffff	Runtime	Coprocessor enables VCX3 (Vector) instructions ([23:16] for VCX3A (Vector), [7:0] for VCX3 (Vector)).
has_cde_tester_trivial	bool	true	true, false	Runtime	Whether the CDETester plugin is implemented (undefs all CDE instructions).

5.7.3 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
```

5.7.4 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 5-6: CDEConstant parameters

Name	Type	Default value	Allowed values	When set	Description
has_cde_constant	bool	true	true, false	Runtime	Whether the CDEConstant plugin is implemented.

5.7.5 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](#) on page 1629

5.7.5.1 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, for example Linux64_GCC7.3.
- A recent version of CMake.
- An installation of the Fast Models package and libraries.

5.7.5.2 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`.

5.7.5.3 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 - parameters](#) on page 1630

5.7.5.4 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`.

5.7.5.5 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 [5.7.1 CDETester](#) on page 1629.

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`.

5.7.5.6 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`.

5.7.6 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`

5.7.6.1 CDEFactoryInterface.h

Defines the interface for obtaining an instance of a `CDEInstHandlerInterface` for a specific core.

5.7.6.1.1 CDE::CDEFactoryInterface class

The factory interface for obtaining an instance of a `CDEInstHandlerInterface` for a specific core.

5.7.6.1.2 CDE::CDEFactoryInterface::instantiateCDEInstHandler()

Instantiate a CDEInstHandlerInterface.

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.

5.7.6.1.3 CDE::CDEFactoryInterface::CDEImplName()

Name of the CDE implementation.

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;
```

5.7.6.1.4 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;
```

5.7.6.1.5 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;
```

5.7.6.2 CDEInstHandlerInterface.h

Defines the interface for executing CDE instructions.

5.7.6.2.1 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.

5.7.6.2.2 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.

5.7.6.2.3 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.

5.7.6.2.4 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.

5.7.6.2.5 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.

5.7.6.2.6 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.

5.7.6.2.7 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.

5.7.6.2.8 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.

5.7.6.2.9 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.

5.7.6.2.10 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.

5.7.6.2.11 CDE::CDEInstHandlerInterface::getCDECoprocessorMask()

Return a bitmask indicating which coprocessor numbers this CDE implementation subsumes.

```
virtual uint8_t getCDECoprocessorMask() = 0;
```

5.7.6.3 CDERegistryInterface.h

Defines the interface to allow components, for instance plug-ins, to contribute CDE implementations to the simulation.

5.7.6.3.1 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.

5.7.6.3.2 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.

5.7.6.3.3 CDE::CDERegistryInterface::unregisterCDEFactory()

Unregister the CDE factory from the simulation.

```
virtual void unregisterCDEFactory(CDEFactoryInterface* interface) = 0;
```

interface

The CDE factory interface used to unregister.

5.7.6.3.4 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;
```

5.8 Crypto

The `crypto` plug-in enables Arm®v8 and Arm®v9 processor models to support the Arm®v8.0 Cryptographic Extensions and Arm®v8.3 architected Pointer Authentication algorithms.

The `crypto` plug-in is available for download from the [Arm Developer website](#).

When the plug-in is loaded:

- All Arm®v8 and Arm®v9 processors in the system implement all functionality from the Arm®v8.0 Cryptographic Extensions by default, although you can disable it by setting the `CRYPTODISABLE` parameter for the core.

- All Arm®v8.3 and Arm®v9 processors in the system implement the architected algorithms for Pointer Authentication and Generic Authentication by default. Plug-in parameters are provided to control which processors in the system have architected algorithms enabled.

AEMs, for example, [4.5.2 AEMvACT](#) on page 196, 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 <code>AESE</code> , <code>AESD</code> , <code>AESMC</code> , and <code>AESIMC</code> instructions are implemented
2	As 1, but in addition, the <code>PMULL</code> and <code>PMULL2</code> 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 <code>SHA1C</code> , <code>SHA1P</code> , <code>SHA1M</code> , <code>SHA1H</code> , <code>SHA1SU0</code> , and <code>SHA1SU1</code> instructions are implemented. This is the default value.

- `crypto_sha256`, with these possible values:

0	No SHA-256 instructions are implemented
1	The <code>SHA256H</code> , <code>SHA256H2</code> , <code>SHA256SU0</code> , and <code>SHA256SU1</code> instructions are implemented. This is the default value.

- `crypto_sha3`, with these possible values:

0	No Arm®v8.4 SHA-3 instructions are implemented. This is the default value.
1	SHA-3 instructions are implemented if Arm®v8.4 is enabled.
2	SHA-3 instructions are implemented.

- `crypto_sha512`, with these possible values:

0	No Arm®v8.4 SHA-512 instructions are implemented. This is the default value.
1	SHA-512 instructions are implemented if Arm®v8.4 is enabled.
2	SHA-512 instructions are implemented.

- `crypto_sm3`, with these possible values:

0	No Arm®v8.4 SM-3 instructions are implemented. This is the default value.
1	SM-3 instructions are implemented if Arm®v8.4 is enabled.
2	SM-3 instructions are implemented.

- `crypto_sm4`, with these possible values:

- | | |
|---|---|
| 0 | No Arm®v8.4 SM-4 instructions are implemented. This is the default value. |
| 1 | SM-4 instructions are implemented if Arm®v8.4 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 Arm®v8-A and Arm®v9-A models, the behavior is fixed to the default values.

5.8.1 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 5-7: Crypto parameters

Name	Type	Default value	Allowed values	Runtime	Description
authentication_algorithm	string	"QARMA5"	""	false	Choice of PACAlgorithm. Valid values: "QARMA5", "QARMA3". "QARMA3" can be enabled only if the core feature has <code>_qarma3_pac</code> is true. Default value: "QARMA5". The pattern is a comma separated list of glob-style patterns. These patterns are matched against the instance name of cores in the system. If a pattern matches then the Architected algorithm is installed on the core if that core supports ARMv8.3. For example: - "*", install the algorithm on all supported cores - "cluster0.*", install the algorithm on all cores in cluster0 - "cluster0.cpu0" install the algorithm only on cluster0.cpu0 - "cluster0.*,cluster1.cpu1" install the algorithm on all cores in cluster0 and also cluster1.cpu1 - "" (an empty string), do not install the algorithm on any core .
generic_authentication_core_pattern	string	"*"	""	false	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 .

Name	Type	Default value	Allowed values	Runtime	Description
pointer_authentication_core_pattern	string	"*"	""	false	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	false	verbosity level. 0, terse. 1, verbose.

5.9 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.

It 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.

5.9.1 GDBRemoteConnection parameters

This section describes the parameters for the `GDBRemoteConnection` plug-in.

Each parameter is prefixed with `REMOTE_CONNECTION.GDBRemoteConnection`, for example:

```
REMOTE_CONNECTION.GDBRemoteConnection.memory_access_size
```

Table 5-8: GDBRemoteConnection parameters

Name	Type	Default value	Allowed values	Runtime	Description
<code>core_selection</code>	int	0	0-128	false	The core to drive with GDB.
<code>listen_address</code>	string	"0.0.0.0"	-	false	Network address the server should listen on.
<code>memory_access_size</code>	int	1	1-8	false	The access size for memory reads/writes in bytes.
<code>port</code>	int	31627	1-65535	false	TCP port the server should listen on.
<code>shutdown_on_disconnect</code>	bool	false	true, false	false	Shut down the model when the client disconnects.

5.9.2 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.
- Memory view only shows the current memory space.
- 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.

5.10 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 must 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
```

Alternatively, 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/`.

5.10.1 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 5-9: GenericCounter parameters

Name	Type	Default value	Allowed values	When set	Description
<code>print_on_event</code>	string	""	""	Runtime	If set, print the count information to stdout when <code>print_on_event</code> trace source fires. If empty, only print the count information at the end of the simulation or when the <code>print_stats</code> parameter is written to.
<code>print_stats</code>	int	0x0	0x0 - 0x0	Runtime	On write, print count information to stdout.
<code>trace-source</code>	string	"INST"	""	Runtime	The trace source to be counted. Example: <code>BRA_DIR</code> .

5.11 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 as a parameter to the plug-in, 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 [5.13 ListTraceSources](#) on page 1651 for details.

The `trace-sources` parameter provides flexibility when specifying trace sources:

- To specify trace sources that match a pattern, use `*` 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 from 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
```

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 [5.20 ToggleMTIPlugin](#) on page 1685.

5.11.1 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 5-10: 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	0
opc2	0

5.11.2 GenericTrace - parameters

This section describes the parameters for the GenericTrace plug-in.

Each parameter is prefixed with `TRACE.GenericTrace`, for example:

```
TRACE.GenericTrace.enabled
```

Table 5-11: GenericTrace parameters

Name	Type	Default value	Allowed values	When set	Description
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.
perf-period	int	0x0	0x0 - 0x7fffffffffffffff	Runtime	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.

Name	Type	Default value	Allowed values	When set	Description
start-icount	int	0x0	0x0 - 0x7fffffffffffffffff	Runtime	Start tracing on a certain instruction count. Tracing starts up to 2048 instructions before this count.
stop-icount	int	0x7fffffffffffffffff	0x0 - 0x7fffffffffffffffff	Runtime	Stop tracing on a certain instruction count. Tracing stops up to 2048 instructions after this count.
stop_on_event	bool	false	true, false	Runtime	Stop the simulation when any event is triggered.
trace-file	string	""	""	Runtime	The trace file to write into. If STDERR, prints to stderr. If empty, prints to stdout.
trace-file-limit	int	0x0	0x0 - 0x7fffffffffffffffff	Runtime	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.

5.12 libete-plugin

The libete-plugin enables the Embedded Trace Extension (ETE) for AEMvA and applicable CPU implementations.



In Fast Models 11.18 and later, ETE support is integrated into the CTModel. We recommend you configure ETE using the model parameters that are prefixed with `ete.`, which are equivalent to the plug-in parameters. The plug-in will be removed in a future release.

5.12.1 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 5-12: libete-plugin parameters

Name	Type	Default value	Allowed values	When set	Description
ASYNC_PACKETS_WHEN_VIEWINST_OFF	bool	false	true, false	Runtime	Generate the non-periodic alignment synchronisation packet generation when trace unit is operative.
ATBTRIG	bool	true	true, false	Runtime	ATB trigger support.
CCITMIN	int	0x4	0x0 - 0xffff	Runtime	Minimum cycle count value.
CCSIZE	int	0xc	0xc - 0x14	Runtime	Cycle counter size.
CLAIMTAGS	int	0x8	0x0 - 0x20	Runtime	Number of claim tags.
COMMOPT	bool	true	true, false	Runtime	Commit mode.
COMMTRANS	bool	false	true, false	Runtime	Commit transaction mode.
DEBUG	int	0x2	0x0 - 0xf	Runtime	DEBUG.
DESIGNER	int	0x41	0x0 - 0xff	Runtime	DESIGNER value.
ETE_REVISION	int	0x0	0x0 - 0x3	Runtime	ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.
EXCEPTION_WITH_CONTEXT	bool	true	true, false	Runtime	Whether EXCEPTION_WITH_CONTEXT packet is supported.
EXPLICITLY_COMMIT_PO_ELEMS	bool	false	true, false	Runtime	Whether to unilaterally explicitly emit a commit after a PO packet.
IMPDEFEXCEPPERCENTAGE	int	0x0	0x0 - 0x32	Runtime	Percentage of IMPDEF exceptions inserted in instruction blocks.
IMPDEF_TRACE_ON	int	0x0	0x0 - 0x3	Runtime	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	Runtime	Number of instruction blocks traced on a transition in the filtering.
LPOVERRIDE	bool	true	true, false	Runtime	Low power override.
MAXSPEC	int	0x0	0x0 - 0xffffffff	Runtime	Maximum speculation depth.
MAX_INST_PER_Q	int	0x1	0x1 - 0x1000	Runtime	Maximum limit for the number of instructions implied by a Q element.
NOOVERFLOW	bool	false	true, false	Runtime	No overflow.
NUMACPAIRS	int	0x4	0x0 - 0x8	Runtime	Number of instruction address comparators pairs.
NUMCIDC	int	0x1	0x0 - 0x8	Runtime	Number of context ID comparators.
NUMCNTR	int	0x2	0x0 - 0x4	Runtime	Number of counters.
NUMEXTINSEL	int	0x4	0x0 - 0x4	Runtime	Number of external input selectors.
NUMPC	int	0x0	0x0 - 0x8	Runtime	Number of PE comparators.
NUMSEQSTATE	int	0x4	0x0 - 0x4	Runtime	Number of sequencer states.
NUMSSCC	int	0x1	0x0 - 0x8	Runtime	Number of single shot comparators.
NUMVMIDC	int	0x1	0x0 - 0x8	Runtime	Number of virtual ID comparators.
NumberOfETEEEvents	int	0x2	0x0 - 0x4	Runtime	Number of trace events.
NumberOfRSPairs	int	0x8	0x0 - 0x10	Runtime	Number of resource selector pairs.
PIDR_CM0D	int	0x0	0x0 - 0xf	Runtime	TRCPIDR CM0D value.
PIDR_DESIGNER	int	0x0	0x0 - 0x7ff	Runtime	TRCPIDR DESIGNER value.

Name	Type	Default value	Allowed values	When set	Description
PIDR_PART	int	0x0	0x0 - 0xffff	Runtime	TRCPIDR PART number value.
PIDR_REVAND	int	0x0	0x0 - 0xf	Runtime	TRCPIDR REVAND value.
PIDR_REVISION	int	0x0	0x0 - 0xf	Runtime	TRCPIDR REVISION value.
QFILT	bool	false	true, false	Runtime	Q filtering.
QSUP	int	0x0	0x0 - 0x3	Runtime	Q support.
Q_CADENCE	int	0x1	0x1 - 0x1000	Runtime	Number of instruction blocks traced between two Q elements.
REG_ACCESS_ONLY_MODE	bool	false	true, false	Runtime	If enabled, all traces are disabled. Plugin only allows register acceses.
RES0_STATEFUL	bool	false	true, false	Runtime	Whether RES0 bits are stateful or RAZ/WI.
RETSTACK	int	0x3	0x0 - 0xf	Runtime	Return stack depth.
REVISION	int	0x0	0x0 - 0xf	Runtime	TRCIDR1 revision value.
SIM_OVERFLOW_GRANULARITY	int	0x64	0xa - 0xffffffff	Runtime	Number of instruction blocks in each granule, for simulated overflow.
SIM_OVERFLOW_PERCENTAGE	int	0x0	0x0 - 0x63	Runtime	Percentage of instruction blocks lost in each granule, for simulated overflow.
SOURCE_ADDRESS	bool	false	true, false	Runtime	Allow generation of source address elements.
STALLCTRL	bool	true	true, false	Runtime	Stall control.
SYSSTALL	bool	true	true, false	Runtime	System stall.
TRACEIDSIZE	int	0x7	0x0 - 0x7	Runtime	Trace ID size.
TRACE_OUTPUT	string	""	""	Runtime	File to which to write trace byte stream.
TRACE_OUTPUT_ENABLE	bool	false	true, false	Runtime	ETE Trace output enable: 1=enable, 0=disable.
TRCRSRTA_FORCED_EXCEP	bool	false	true, false	Runtime	TRCRSR.TA value for a forcibly traced exception.
TSMARK	bool	false	true, false	Runtime	Whether timestamp markers are supported.
TSSIZE	int	0x8	0x8 - 0x8	Runtime	Timestamp size.
WFXMODE	bool	true	true, false	Runtime	WFX mode.

5.13 ListTraceSources

ListTraceSources is an MTI plug-in that displays a complete and self-documenting list of the trace sources that a model provides, without running the model.

The plug-in prints output for each component in the model, either to stdout or to a file, for example:

```
...
Component (4) providing trace: FVP_VE_Cortex_A7x1.cluster.cpu0 (ARM_Cortex-A7,
11.4.60)
=====
Component is of type "ARM_Cortex-A7"
Version is "11.4.60"
#Sources: 195
```

```
Source ASYNC_MEMORY_FAULT (Context ID Register write.)
  Field FAULT type:MTI_UNSIGNED_INT size:4 (Fault status in ESR format)
  Field VADDR type:MTI_SIGNED_INT size:8 (Virtual Address (or 0 if unavailable))
  Field PADDR type:MTI_UNSIGNED_INT size:8 (Physical Address (or 0 if
unavailable))
...
```

The source code for this plug-in is also provided as a programming example in `$PVLIB_HOME/examples/MTI/ListTraceSources/source/`.

5.13.1 ListTraceSources - parameters

This section describes the parameters for the ListTraceSources plug-in.

Each parameter is prefixed with `TRACE.ListTraceSources`, for example:

```
TRACE.ListTraceSources.file
```

Table 5-13: ListTraceSources parameters

Name	Type	Default value	Allowed values	When set	Description
file	string	""	""	Runtime	File to write the list of trace sources to. Default is to write to the console.
print_components_only	bool	false	true, false	Runtime	If true, the plug-in prints the trace component information only, not the sources or fields.

5.14 PipelineModel

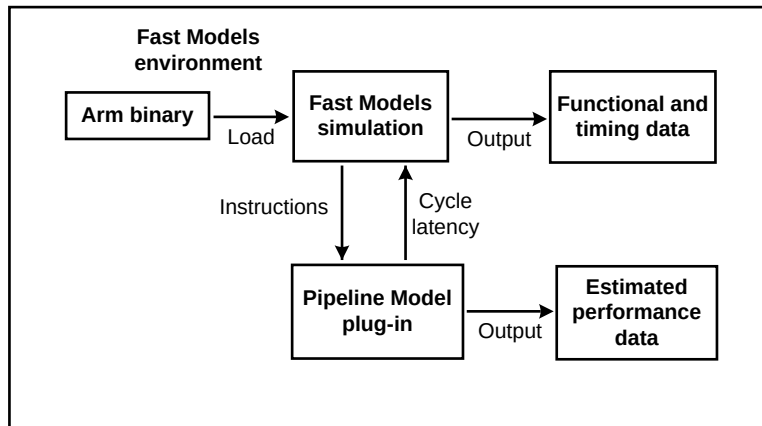
Use the PipelineModel plug-in to estimate the performance of workloads within a Fast Models environment.



The PipelineModel plug-in is deprecated. It might be modified or removed in a future release.

The plug-in models first-order effects of microarchitecture components on the overall Cycles Per Instruction (CPI) value. Examples of such effects are data and structural hazards due to instruction dependencies.

The PipelineModel is implemented as a Fast Models plug-in. It processes instruction traces and injects cycle latencies into the simulation. The plug-in is non-intrusive to the functional accuracy of the simulation.

Figure 5-1: PipelineModel plug-in overview

Fast Models provides the following prebuilt example PipelineModel plugins:

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.



Note

The model should not be used as a reference for hardware performance as it has limitations, such as the absence of a cache model.

InOrderPipelineModel

An implementation of a single-issue 4-stage pipelined processor that illustrates the basic components of a CPU. It demonstrates how components such as Fetch, Decode, and Execute can be implemented as a Fast Models plug-in.

The source code is provided in `$PVLIB_HOME/plugins/source/PipelineModel/Cores/InOrder/`. It contains a README and a makefile for building the example.



Note

The model does not represent any Arm® core and is intended only as a guide for developing more advanced PipelineModels using Fast Models.

The `PipelineModel` is an aspect of Timing Annotation. For more details, see [Timing Annotation](#) in the *Fast Models User Guide*.

5.14.1 PipelineModel parameters

This section describes the parameters for the `CortexA53PipelineModel` plug-in example. Other `PipelineModel` plug-ins might have different parameters.

Each parameter is prefixed with `PipelineModel` and the plug-in name, or plug-in instance name. For example:

```
PipelineModel.CortexA53PipelineModel.core-type
```

Table 5-14: PipelineModel parameters

Name	Type	Default value	Description
<code>core-type</code>	string	""	The name of the core. This value should match the platform that the plug-in is being attached to. For example: <code>ARM_Cortex-A53</code> . This parameter is required.
<code>instance-name</code>	string	""	The cluster and core instance name, for example: <code>cluster0.cpu0</code> . This parameter is optional. If it is specified, the plug-in only attaches to the specified instance. If it is not specified, the plug-in attaches to any instance.
<code>output</code>	string	""	The path and filename of the output statistics file, or <code>"stderr"</code> . This parameter is optional. If it is not specified, output is sent to <code>stdout</code> .
<code>start-pc</code>	string	"0x0"	The <code>PipelineModel</code> starts processing instructions from this PC value. If you want the <code>PipelineModel</code> to process only a subset of a program, use the <code>start-pc</code> and <code>end-pc</code> parameters. These values might represent the entry and exit of a function call, for example. After the <code>start-pc</code> address is hit, the <code>PipelineModel</code> processes instructions until one of the following occurs: <ul style="list-style-type: none"> <code>end-pc</code> is hit. A cycle limit, specified using the <code>--cyclelimit</code> model option, is reached. The simulation is terminated. If <code>end-pc</code> is hit, the <code>PipelineModel</code> stops processing instructions and only resumes if <code>start-pc</code> is hit again. This parameter is optional. If it is not specified, or if <code>start-pc</code> and <code>end-pc</code> are the same, the plug-in processes all instructions.
<code>end-pc</code>	string	"0x0"	The <code>PipelineModel</code> stops processing instructions when this PC value is reached. This parameter is optional. If it is not specified, or if <code>start-pc</code> and <code>end-pc</code> are the same, the plug-in processes all instructions.

5.14.2 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.

5.14.3 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
```

5.14.4 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
```

Multicluster, 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=8000FFFF
```

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

It might be useful to run the `PipelineModel` with a cycle limit. You can do this using 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` will not process any instructions.

5.14.5 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, where:

$$CPI = \text{cycles elapsed} / \text{instructions retired}$$

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:

$$IPC = \text{instructions retired} / \text{cycles elapsed}$$

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.

5.15 RunTimeParameterTest

`RunTimeParameterTest` is an example MTI plug-in that demonstrates how to add new string, integer, and boolean parameters at runtime.

This plug-in is provided only as source code, in `$PVLIB_HOME/examples/MTI/RunTimeParameterTest/source/`.

5.16 ScalableVectorExtension

The `ScalableVectorExtension` plug-in enables AEMvA models to support the Scalable Vector Extension (SVE) and SVE2.



Note

- In Fast Models 11.24 and later, support for SVE and SVE2 is built into CT models. This built-in support gives behavior that matches the TRM. If you use the plug-in with one of these models, the plug-in overrides the built-in behavior. This plug-in will be removed in a future release.
- To enable or disable support for particular features in the plug-in, use the `has_*` parameters.
- SVE and SVE2 are optional extensions to the Arm® architecture. For information about SVE and SVE2, see [Introduction to SVE](#) and [Introducing SVE2](#).

5.16.1 ScalableVectorExtension - parameters

This section describes the parameters for the `ScalableVectorExtension` plug-in.

Each parameter is prefixed with `SVE.ScalableVectorExtension`, for example:

```
SVE.ScalableVectorExtension.clear_constrained_lanes
```

Table 5-15: ScalableVectorExtension parameters

Name	Type	Default value	Allowed values	When set	Description
<code>clear_constrained_lanes</code>	int	0x0	0x0 - 0x2	Runtime	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.
<code>combine_movprfx_and_destructive</code>	bool	false	true, false	Runtime	Attempt to combine the execution of MOVPRFX and the destructively-encoded instruction that follows it.
<code>disable_speculative_accesses</code>	bool	false	true, false	Runtime	All speculative memory accesses behave as though faulting, without accessing memory.
<code>enable_at_reset</code>	bool	false	true, false	Runtime	Start with system registers set up for Scalable Vector Extension use.
<code>ffr_16b_pattern_UNKNOWN</code>	int	0x0	0x0 - 0xffff	Runtime	A specific 16-bit UNKNOWN value that is used by parameter <code>force_UNKNOWN_to_ffr</code> .
<code>force_UNKNOWN_to_ffr</code>	int	0x0	0x0 - 0x3	Runtime	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 <code>ffr_16b_pattern_UNKNOWN</code> , 2 - Clear all bits above first zero 3 - Set all bits after first one.

Name	Type	Default value	Allowed values	When set	Description
fp_exception_report_lowest	bool	false	true, false	Runtime	If true, for multiple trapped FP exceptions, report the lowest lane in VECITR. Otherwise, report the highest.
fp_exception_set_tfv	bool	true	true, false	Runtime	Set ESR_ELx.TFV during FP exception. Trapped exception flags are valid.
fp_exception_set_vecitr	bool	false	true, false	Runtime	If true, set ESR_ELx.VECITR during FP exception. Otherwise, set RES0.
has_b16b16	int	0x1	0x0 - 0x1	Runtime	Whether FEAT_B16B16 is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if either FEAT_SVE2p1 or FEAT_SME2p1 is implemented.
has_sme	bool	false	true, false	Runtime	Whether SME is implemented.
has_sme2	bool	false	true, false	Runtime	Whether SME2 is implemented (FEAT_SME2).
has_sme_f16f16	int	0x1	0x0 - 0x1	Runtime	Whether FEAT_SME_F16F16 is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if FEAT_SME2p1 is implemented.
has_sme_f64f64	int	0x1	0x0 - 0x1	Runtime	If SME is implemented, whether double-precision FMOPA and FMOPS are implemented.
has_sme_f8f16	int	0x1	0x0 - 0x1	Runtime	If SME2 is implemented, whether FEAT_SME_F8F16 is implemented.
has_sme_f8f32	int	0x1	0x0 - 0x1	Runtime	If SME2 is implemented, whether FEAT_SME_F8F32 is implemented.
has_sme_fa64	bool	false	true, false	Runtime	Whether FEAT_SME_FA64 is implemented.
has_sme_i16i64	int	0x1	0x0 - 0x1	Runtime	If SME is implemented, whether instructions that accumulate 16-bit integer outer products into 64-bit integer tiles are implemented.
has_sme_lutv2	bool	false	true, false	Runtime	Whether FEAT_SME_LUTv2 is implemented.
has_sme_priority_control	bool	true	true, false	Runtime	Whether SME Priority Control is implemented.
has_sve2	bool	false	true, false	Runtime	Whether SVE2 is implemented (FEAT_SVE2).
has_sve2_aes	int	0x2	0x0 - 0x2	Runtime	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 (FEAT_SVE_PMULL128).
has_sve2_bit_perm	bool	true	true, false	Runtime	If SVE2 is implemented, whether BitPerm instructions are implemented (FEAT_SVE_BitPerm).
has_sve2_sha3	bool	true	true, false	Runtime	If SVE2 is implemented, whether SHA3 instructions are implemented (FEAT_SVE_SHA3).
has_sve2_sm4	bool	true	true, false	Runtime	If SVE2 is implemented, whether SM4 instructions are implemented (FEAT_SVE_SM4).

Name	Type	Default value	Allowed values	When set	Description
has_sve_bf16	bool	true	true, false	Runtime	Whether SVE BFloat16 instructions are implemented.
has_sve_extended_bf16	int	0x2	0x0 - 0x2	Runtime	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.
has_sve_mm_f32	bool	true	true, false	Runtime	Whether the SVE FP32 Matrix Multiply instructions are implemented (FEAT_F32MM).
has_sve_mm_f64	bool	true	true, false	Runtime	Whether the SVE FP64 Matrix Multiply instructions are implemented (FEAT_F64MM).
has_sve_mm_i8	bool	true	true, false	Runtime	Whether the SVE Int8 Matrix Multiply instructions are implemented (FEAT_I8MM).
movprfx_unpredictable_behavior	int	0x0	0x0 - 0x3	Runtime	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.
predicated_sp_align_check_behaviour	int	0x0	0x0 - 0x3	Runtime	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.
relax_sme_watchpoint_matching_16	bool	false	true, false	Runtime	Whether memory accesses through Z and P registers in Streaming Mode and all accesses through ZA match watchpoints rounded to 16-byte alignment.
relax_sve_watchpoint_matching_16	bool	false	true, false	Runtime	If FEAT_DEBUGv8p9 is implemented, whether memory accesses through Z and P registers outside Streaming Mode match watchpoints rounded to 16-byte alignment.
sm_tag_checked	bool	true	true, false	Runtime	Whether SME, SVE, and SIMD&FP load and store instructions executed when the PE is in Streaming SVE mode perform a Tag Check.
sme2_version	int	0x0	0x0 - 0x1	Runtime	The version of SME2 if implemented. Possible values are: 0 - FEAT_SME2, 1 - FEAT_SME2p1.
sme_only	bool	false	true, false	Runtime	If SME is implemented, whether SVE functionality is available only when SM=1.
sme_ssve_fp8_support_level	int	0x0	0x0 - 0x3	Runtime	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_F8FMA, 2 - FEAT_SSVE_F8DOT4, 3 - FEAT_SSVE_F8DOT2.
sme_veclens_implemented	int	0x7	0x1 - 0x1f	Runtime	Which SME vector lengths are implemented. Represented as a bitfield where bit[n]==1 implies SME vector length of $128 \cdot 2^n$ bits is implemented.
smidr_el1_implementation_val	int	0x41	0x0 - 0xff	Runtime	The value of SMIDR_EL1.Implementer.

Name	Type	Default value	Allowed values	When set	Description
smidr_el1_revision_val	int	0x0	0x0 - 0xff	Runtime	The value of SMIDR_EL1.Revision.
sve2_version	int	0x0	0x0 - 0x1	Runtime	The version of SVE2 if implemented. Possible values are: 0 - FEAT_SVE2, 1 - FEAT_SVE2p1.
sve_dabt_far_behaviour	int	0x0	0x0 - 0x2	Runtime	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_wp_far_behaviour	int	0x0	0x0 - 0x3	Runtime	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.
trace_za_tilewise	bool	true	true, false	Runtime	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.
undef_invalid_combined_movprfx	bool	true	true, false	Runtime	If a combined MOVPRFX is invalid, raise an UNDEF exception. Otherwise, NOP the second half. This parameter is deprecated.
unknown_value	int	-0x2152215221522153	0x0 - -0x1	Runtime	Simulated value for a state that has an UNKNOWN value after reset.
veclen	int	0x8	0x2 - 0x20	Runtime	SVE vector length in units of 64 bits.
z_reg_on_load_fault_behaviour	int	0x0	0x0 - 0x1	Runtime	Governs the behavior of destination Z-registers in case of a load fault. Possible values are: 0 - Register becomes UNKNOWN, 1 - Register is preserved.
za_on_svl_increase_behaviour	int	0x0	0x0 - 0x1	Runtime	Controls the state of the previously inaccessible portion of the ZA registers on SVL increase. Possible values are: 0 - Retain values, 1 - Zero ZA.
za_tag_checked	bool	true	true, false	Runtime	Whether memory accesses due to SME LDR and STR instructions that access the SME ZA array perform a Tag Check.

5.17 Sidechannel

The Generic Graphics Accelerator uses this plug-in for communication between the target and the host.

For more information, see [Graphics Acceleration in Fast Models](#) in the Fast Models User Guide.

5.17.1 Sidechannel - parameters

This section describes the parameters for the Sidechannel plug-in.

Each parameter is prefixed with `DEBUG.Sidechannel`, for example:

```
DEBUG.Sidechannel.diagnostics
```

Table 5-16: Sidechannel parameters

Name	Type	Default value	Allowed values	When set	Description
diagnostics	int	0x0	0x0 - 0x3	Runtime	Diagnostic level (0=none, 1=calls, 2=with data dump).
interceptor	string	""	""	Runtime	Interceptor .SO to load.
sh-diagnostics	int	0x0	0x0 - 0x3	Runtime	Diagnostic level for semihosting mechanism.

5.18 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 processors 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 if multiple components are given. 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.

5.18.1 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 5-17: TarmacText parameters

Name	Type	Default value	Allowed values	When set	Description
component	string	""	""	Runtime	Space-separated paths of the CPU instances to trace.
log	string	"tarmac"	""	Runtime	Log tarmac text in a file named \$log.@PATH@.log.
start	int	0x0	0x0 - 0x0	Runtime	Start tracing after the specified number of steps. Note: Tracing starts at the end of the quantum in which the count has been reached. Reducing the quantum size reduces the latency but might impact performance.
use_instr_cnt_as_timestamp	int	0x0	0x0 - 0x0	Runtime	Use the instruction count as the timestamp instead of the simulation time.

5.19 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.



Note

This plug-in can be used with [5.20 ToggleMTIPlugin](#) on page 1685.

5.19.1 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 5-18: TarmacTrace parameters

Name	Type	Default value	Allowed values	When set	Description
end-instruction-count	int	0x0	0x0 - 0x7fffffffffffffff	Runtime	The instruction count when the tracing should be stopped. Default is to never stop tracing.
instruction-count-is-per-target	bool	true	true, false	Runtime	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	Runtime	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	Runtime	Set the default quantum size used to compute when the tracing should start and stop, in instructions. This is overridden by the CORE_INFO.QUANTUM_SIZE trace source field of the component, if present.
quiet	bool	false	true, false	Runtime	Limit output to trace information.
start-instruction-count	int	0x0	0x0 - 0x7fffffffffffffff	Runtime	The instruction count when the tracing should start. Default is to trace from the beginning.
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	Runtime	Write trace to multiple files.
trace-inst-stem	string	""	""	Runtime	Base instance path to select a group of instances to trace.
trace_aarch64_vfp_full_width	bool	false	true, false	Runtime	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	Runtime	Trace memory update by atomic operation.
trace_branches	bool	false	true, false	Runtime	Trace changes of the program flow like direct or indirect branches and exception returns.
trace_bte	bool	true	true, false	Runtime	Trace opcode rejected by BTE.
trace_bus_accesses	bool	false	true, false	Runtime	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	Runtime	Trace cache fills and evictions.
trace_core_registers	bool	true	true, false	Runtime	Trace the core registers R0-R14, the CPSR, and the SPSR registers.
trace_cp15	bool	true	true, false	Runtime	Trace writes to CP15 registers.
trace_dap	bool	true	true, false	Runtime	Trace accesses on the debug access port.
trace_ete	bool	true	true, false	Runtime	Trace packets generated by the ETE.
trace_events	bool	true	true, false	Runtime	Trace events, for example exceptions and mode changes.

Name	Type	Default value	Allowed values	When set	Description
trace_exception_reasons	bool	true	true, false	Runtime	Trace INFO_EXCEPTION_REASONS (M-class only so far).
trace_generic_events	bool	false	true, false	Runtime	Trace generic events.
trace_gicv3	bool	true	true, false	Runtime	Trace GICv3 memory mapped accesses.
trace_gicv3_comms	int	0x0	0x0 - 0x7	Runtime	Trace GICv3 communications between cores and distributor. Bitfield; 1 = trace CPU; 2 = trace RDO; 4 = trace internal.
trace_gicv3_its	bool	false	true, false	Runtime	Trace GICv3 ITS command execution.
trace_gicv3_reads	bool	false	true, false	Runtime	Trace GICv3 memory mapped reads.
trace_gpt	bool	true	true, false	Runtime	Trace packets generated by the GPT.
trace_hacdbss	bool	true	true, false	Runtime	Trace packets generated by the HACDBS.
trace_hdbss	bool	true	true, false	Runtime	Trace packets generated by the HDBSS.
trace_instructions	bool	true	true, false	Runtime	Trace instructions.
trace_loads_store_memtype	bool	false	true, false	Runtime	Show memory type information for core loads and stores.
trace_loads_stores	bool	true	true, false	Runtime	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	Runtime	Represent non-updated bytes as ---- in S-registers trace.
trace_memory	bool	false	true, false	Runtime	Trace memory accesses just outside the core.
trace_mmu	bool	true	true, false	Runtime	Trace mmu tablewalks and associated information.
trace_spe	bool	true	true, false	Runtime	Trace SPE data written to memory.
trace_tag_loads_stores	bool	true	true, false	Runtime	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	Runtime	Trace the VFP and Neon registers, including FPSCR and FPEXC.
unbuffered	bool	false	true, false	Runtime	Trace events as they arrive and flush each fwrite. Prints IT even when IS should be printed.
updated-registers	bool	false	true, false	Runtime	Trace the updated value of registers rather than the written value.
use_inst_end_for_inst_counter	bool	false	true, false	Runtime	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	false	true, false	Runtime	Use the instruction count as the timestamp instead of the simulation time.

5.19.2 TarmacTrace file format

This section describes the TarmacTrace file format.



[*x*|*y*]

Indicates a choice between *x* and *y*.

{*x*}

Indicates that *x* is optional or configuration dependent.

The common address definition that is used in the trace command syntax:

```
<vaddr>{:<paddr><psecurity>}
```

<vaddr>

The virtual address in hexadecimal format. See the note below.

<paddr>

The physical address of the instruction in hexadecimal. See the note below. <paddr> is only added if it is different to <vaddr>.

<psecurity>

Append **_ns** if the security regime of the physical address is Non-secure. Append nothing if the regime is Secure.



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.

The virtual regime definition that is used in the trace command syntax:

```
0x<vbase>{_NS} <el>{ vmid=<vmid>}{, nG asid=<asid>}
```

0x<vbase>

Virtual address in hexadecimal format.

_NS

If present, this element specifies that the address is Non-secure. If not present, the address is Secure.

<el>

Translation regime that owns the mapping. One of:

- **EL1_n**, meaning the Non-secure EL1&O translation regime.
- **EL2_n**

- EL1_s
- EL3_s

<vmid>

For Non-secure, non-hyp regimes, specify the VMID.

nG

If present, this element specifies that the virtual regime is non-global.

asid

For non-global regimes, specify the ASID.

5.19.3 Tarmac Trace output example

This example output from the Tarmac Trace plug-in shows various trace types, 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 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
...
```

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 + 0x10` 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 an instruction that was taken. To interpret the values in this line, see [5.19.4 Instruction trace](#) on page 1670. 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 [5.19.12 Processor memory access trace](#) on page 1680. 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. An underscore separates groups of 8 digits.
- 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 [5.19.6 Register trace](#) on page 1672. 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 LPAE 1:1 000080002010 0000000080000705 : BLOCK ATTRIDX=1 NS=0
AP=0 SH=3 AF=1 nG=0 16E=0 PXN=0 XN=0 ADDR=0x0000000080000000
98 clk TLB FILL cpu0.UTLB 1G 0x80000000 NS EL1_n vmid=0:0x0080000000 NS Normal
InnerShareable Inner=WriteBackWriteAlloc Outer=WriteBackWriteAlloc xn=0
pxn=0 ContiguousHint=0 xs=0
```

- `TTW` indicates a translation table walk trace. To interpret the values in this line, see [5.19.9 Translation table walk trace](#) on page 1676. For example:
 - `ITLB` means instruction TLB.
 - `LPAE` means Large Physical Address Extension (LPAE)-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 in hexadecimal.
 - Following the colon is the parsed result. In this case, the LPAE region descriptor.

- `TLB` indicates a TLB trace. To interpret the values in this line, see [5.19.10 TLB trace](#) on page 1677. 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=0:0x0080000000_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.

5.19.4 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 [5.19.2 TarmacTrace file format](#) on page 1667.

<opcode>

16-bit/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.

Related information

[TarmacTrace file format](#) on page 1667

5.19.5 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 [5.19.2 TarmacTrace file format](#) on page 1667.

<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

Related information

[TarmacTrace file format](#) on page 1667

5.19.6 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`, `FPCR`, and `FPEXC`, and writes to system registers including `CP14`, `CP15`, and `GIC`. Banked registers are traced separately using the mode as a suffix to the register name, for example `r13` (current register `R13`) and `r13_mon` (banked register `R13`).

Output syntax:

```
<time> <scale> {<cpu>} R <register> <value>
```

<time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

<scale>

Unit for `<time>`. `ps` means simulation time, `clk` means instruction count.

<cpu>

Processor, or other component, that gave the instruction.

<register>

Register name. Banked core registers can have a mode appended to them with a single underscore. Banked `CP14/CP15` registers have `_s` or `_ns` appended to indicate access of either the Secure or Non-secure banked register.



Note

In Arm®v8 and Arm®v9, when the register name is `cpsr`:

- In AArch64 state, `cpsr` is used to trace PSTATE changes. The bit format of `<value>` follows the `SPSR_ELx` AArch64 format.
- In AArch32 state, the bit format of `<value>` follows the `CPSR` format.

<value>

Hexadecimal value that is written to the register (64 bits maximum).

If the SVE plug-in is loaded in the model, there are additional registers in the program view. The output examples below show how these registers are traced when the value changes. These data values can be very large.

```
8463 clk cpu0 IT (8439) 000282c0:0000152282c0 NS 053fc01f O EL1h_n : SEL z31.B,p0,z0.B,z31.B
8463 clk cpu0 R z31 00000000_00000000_00000000_00000000
```

`R` indicates a register write. `z0` to `z31` are the vector registers. The written data are hexadecimal digits, which are separated by an underscore every 32 bits. The length of the written data varies with the configuration, depending on the vector length.

```
9756 clk cpu0 IT (9732) 01000074:000011000074 NS 2518e3e0 O EL1t_n : PTRUE p0.B,ALL
```

```
9756 clk cpu0 R p0 ffff
```

R indicates a register write. p0 to p15 are the predicate registers. The written data are hexadecimal digits. If they are long enough to require one, the digits are separated by an underscore every 32 bits. The length of the written data varies with the configuration, depending on the vector length. Predicate registers contain 1 bit per byte of vector register length.

5.19.7 Cache maintenance trace

If enabled, this source traces all cache maintenance operations that the processor initiates.

Output syntax:

```
<time> <scale> <cpu> CACHE MAINTENANCE <side> <operation> <scope> <data> {<pagesize> <memtype>}
```

<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.

<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 [5.19.2 TarmacTrace file format](#) on page 1667, 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 which contains the MVA.

<memtype>

If the operation is by MVA, this element is the type of memory in the TLB entry which contains the MVA.

Related information

[TarmacTrace file format](#) on page 1667

5.19.8 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>. `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.

5.19.9 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.

LARGEPAGE {<attr>=<value>}

VMSA region descriptor.

SMALLPAGE {<attr>=<value>}

VMSA region descriptor.

5.19.10 TLB trace

If enabled, this source traces TLB entries that are filled and evicted by the processor.

Output syntax:

```
<time> <scale> <cpu> [TLB|WALKCACHE] FILL <id> <size> <virtualregime>:<paddr>
{<mentype>} {<attr>=<value>}+
```

```
<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 [5.19.2 TarmacTrace file format](#) on page 1667.

<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.

Related information

[TarmacTrace file format](#) on page 1667

5.19.11 Event trace

If enabled, this source 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 [5.19.2 TarmacTrace file format](#) on page 1667.

<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.

Table 5-19: 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	-

Number	Event description	Value
0x00000081	CoreEvent_CURRENT_SPO_IRQ	-
0x00000082	CoreEvent_CURRENT_SPO_FIQ	-
0x00000083	CoreEvent_CURRENT_SPO_ABORT	-
0x00000084	CoreEvent_CURRENT_SPx_SYNC	-
0x00000085	CoreEvent_CURRENT_SPx_IRQ	-
0x00000086	CoreEvent_CURRENT_SPx_FIQ	-
0x00000087	CoreEvent_CURRENT_SPx_ABORT	-
0x00000088	CoreEvent_LOWER_64_SYNC	-
0x00000089	CoreEvent_LOWER_64_IRQ	-
0x0000008A	CoreEvent_LOWER_64_FIQ	-
0x0000008B	CoreEvent_LOWER_64_ABORT	-
0x0000008C	CoreEvent_LOWER_32_SYNC	-
0x0000008D	CoreEvent_LOWER_32_IRQ	-
0x0000008E	CoreEvent_LOWER_32_FIQ	-
0x0000008F	CoreEvent_LOWER_32_ABORT	-



Note

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.

Related information

[TarmacTrace file format](#) on page 1667

5.19.12 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, cik 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 *a*. *_cas_c* shows the value to compare and *_cas_a* 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 *mv* trace source.

<addr>

Virtual address that is used to access memory. Formatted according to the common address definition, see [5.19.2 TarmacTrace file format](#) on page 1667.

<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).

5.19.13 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 [5.19.2 TarmacTrace file format](#) on page 1667.

<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).

5.19.14 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.

5.20 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 slow down the simulation and result in very large trace files. ToggleMTIPlugin helps to avoid these problems by enabling you to toggle trace generation during the simulation. Toggling means that if tracing is on, it is turned off, and vice versa.

ToggleMTIPlugin can be used with the following plug-ins:

- ASTFplugin
- GenericTrace
- TarmacTrace

5.20.1 How to use ToggleMTIPlugin

As with other plug-ins, load ToggleMTIPlugin using the `--plugin` command-line option when launching the model.



- 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. To control which one to use, set the value of the `use_hlt` plug-in parameter:

- `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.



ASTFplugin creates files with 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 ToggleMTIPlugin is used. 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.

5.20.2 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 5-20: ToggleMTIPlugin parameters

Name	Type	Default value	Allowed values	When set	Description
diagnostics	bool	false	true, false	Runtime	Print diagnostics.
disable_mti_from_start	bool	false	true, false	Runtime	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	Runtime	16-bit integer used in HLT instruction meant to be used by this plugin.
use_hlt	bool	true	true, false	Runtime	If true, use HLT #imm16 instruction to toggle MTI behavior.

6. Fast Models examples

The following top-level example directories are installed in `$PVLIB_HOME/examples`.

Table 6-1: 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.

6.1 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 6-2: 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](#)

6.2 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 [6.3 Build and run an FVP example](#) on page 1690.

The following LISA examples are provided:

Table 6-3: 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 7. Base Platform: Platform and Components on page 1698.
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 7.2 Base Platform RevC on page 1699.
FVP_Coproc_Demo	Example implementation of the Coprocessor interface. Registers the coprocessor with a ARMCortexM33CT or ARMAEMv8MCT model. For more information, see 3.6.1 CoprocBusProtocol protocol on page 96.
FVP_MPS2	Source and project files for MPS2-based example platforms. For information about the MPS2 platforms, see 8. Microcontroller Prototyping System 2 on page 1728.
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 9. Versatile Express Model: Platform and Components on page 1740.
VP_PChannel	Shows how to create power controllers to control the power state of the cores and cluster, using the PChannel protocol. For information about PChannel, see 3.5.1 PChannel protocol on page 95.

6.3 Build and run an FVP example

The FVP examples are located under `$PVLIB_HOME/examples/LISA/`.

Before you begin

Ensure the following:

- You have installed Fast Models and SystemC 2.3.4 and have set the environment variables as described in [Installation](#) in the *Fast Models User Guide*.
- You have set up a Fast Models license, either using FlexNet Licensing or [User-Based Licensing](#).
- You are using a supported Operating System and have set up a compatible toolchain, from those listed in [Requirements for Fast Models](#) in the *Fast Models User Guide*.

About this task

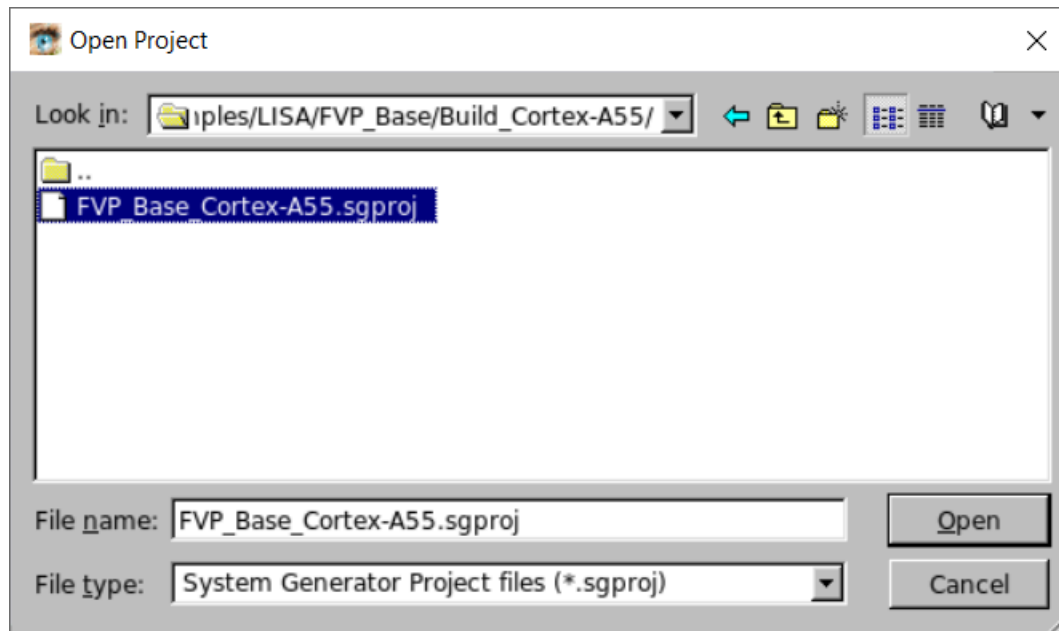
Follow these instructions to build and run one of the Base Platform FVP examples.



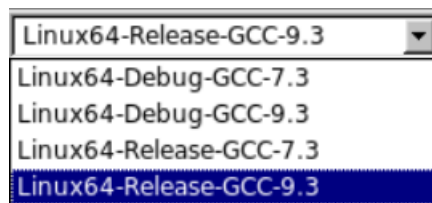
These examples build an executable platform model, which is sometimes referred to as an ISIM (Integrated SIMulator).

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 6-1: 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 6-2: 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/brot_ve_64.axf \  
-C bp.secure_memory=0
```

where:

-a

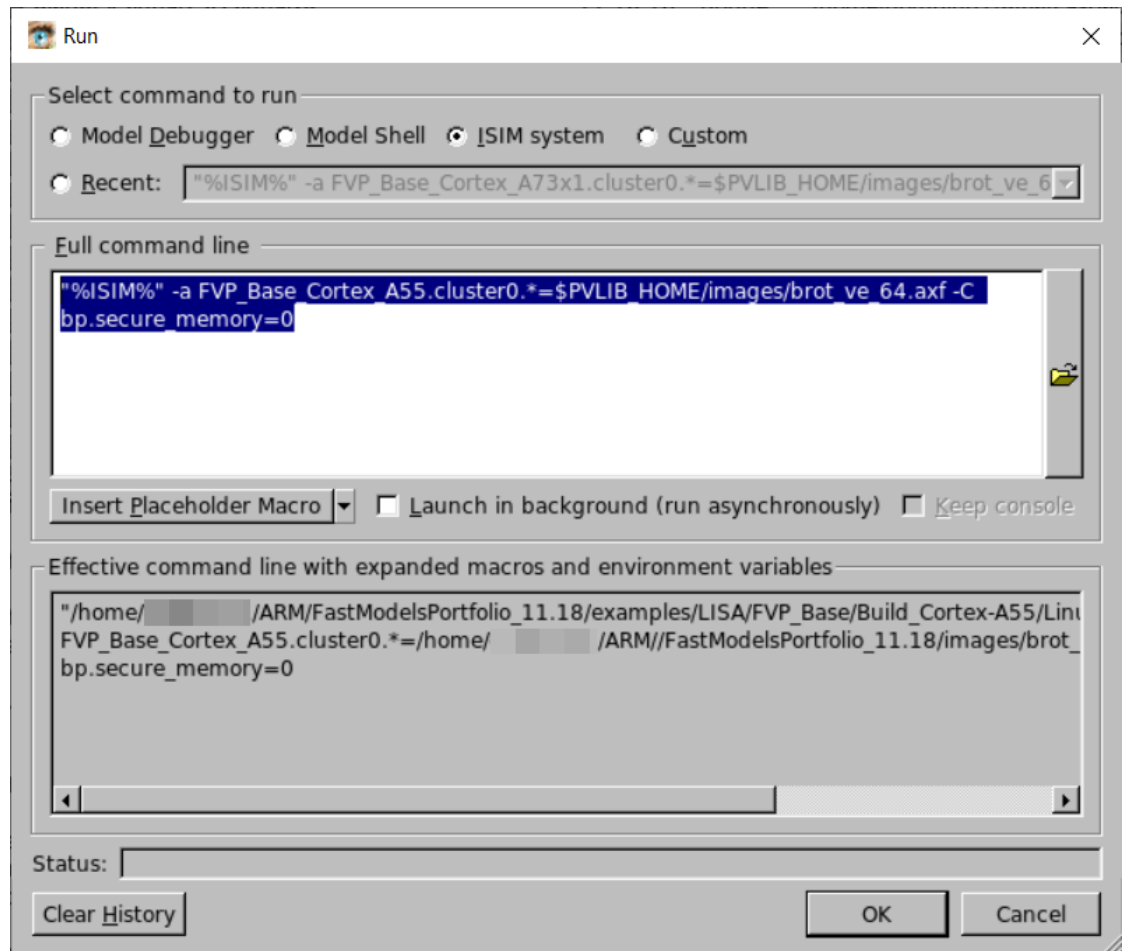
is the name of the application to load. Optionally it also specifies which core instances to run it on, in this case all cores in cluster0.

-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/  
brot_ve_64.axf -C bp.secure_memory=0
```


Figure 6-3: 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 User Guide* for information on modifying, rebuilding, and debugging the example.

Related information

[System Canvas tutorial](#)

[Fast Models glossary](#)

[LISA examples](#) on page 1689

6.4 LISAPlus examples

Source code and System Canvas project files for some example LISA+ components.

Table 6-4: 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 137

[PVBusMaster](#) on page 138

[PVBusSlave](#) on page 139

6.5 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 6-5: 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_EL0 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 5.10 GenericCounter on page 1645.
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 5.11 GenericTrace on page 1646.
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 5.13 ListTraceSources on page 1651.
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](#)

6.6 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 6-6: SystemCExport examples

Directory	Description
Bridges	LISA+ source for bridge components. For more information, see 4.2 Bridge components on page 106.
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 an SVP. Build the component to generate the header file and libraries required to include it in a SystemC simulation.
EVS_Platforms	LISA+ source and makefiles for EVS platform examples. An EVS platform is a LISA+ platform that has been exported as a SystemC object to allow it to be integrated into a SystemC simulation. The EVS platform examples are minimal platforms that are designed for a specific use case, for example running the Dhrystone benchmark application or booting Linux. The Dhrystone 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> . For more information about building EVS platforms, see 6.7 Build and run an EVS platform example on page 1696.

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>

6.7 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/brot_ve_64.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 ISIM 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 User Guide* for information on modifying, rebuilding, and debugging the example.

Related information

[SystemCExport examples](#) on page 1695

7. Base Platform

This chapter describes the Base Platform system model.

7.1 About the 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 standalone products and as examples in Fast Models.

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](#) on page 1689

7.2 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 [4.10.77 SMMUv3AEM](#) on page 1581 with a control region at 0x002B400000 to 0x002B4FFFFFFF.
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 0x002B5FFFFFFF.
- A CoreLink™ [4.10.6 CCI550](#) on page 1201 Cache Coherent Interconnect. Base Platform has a CCI-400.
- An AHCI controller model including a SATA disk model.
- An [4.7.36 SMMUv3TestEngine](#) on page 1167 component with a control region at 0x002bfe0000 to 0x002bffFFFFFF. 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 [4.6.17 Mali_G76](#) on page 1127 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 [Arm Architecture Models](#) on Arm Developer.

Related information

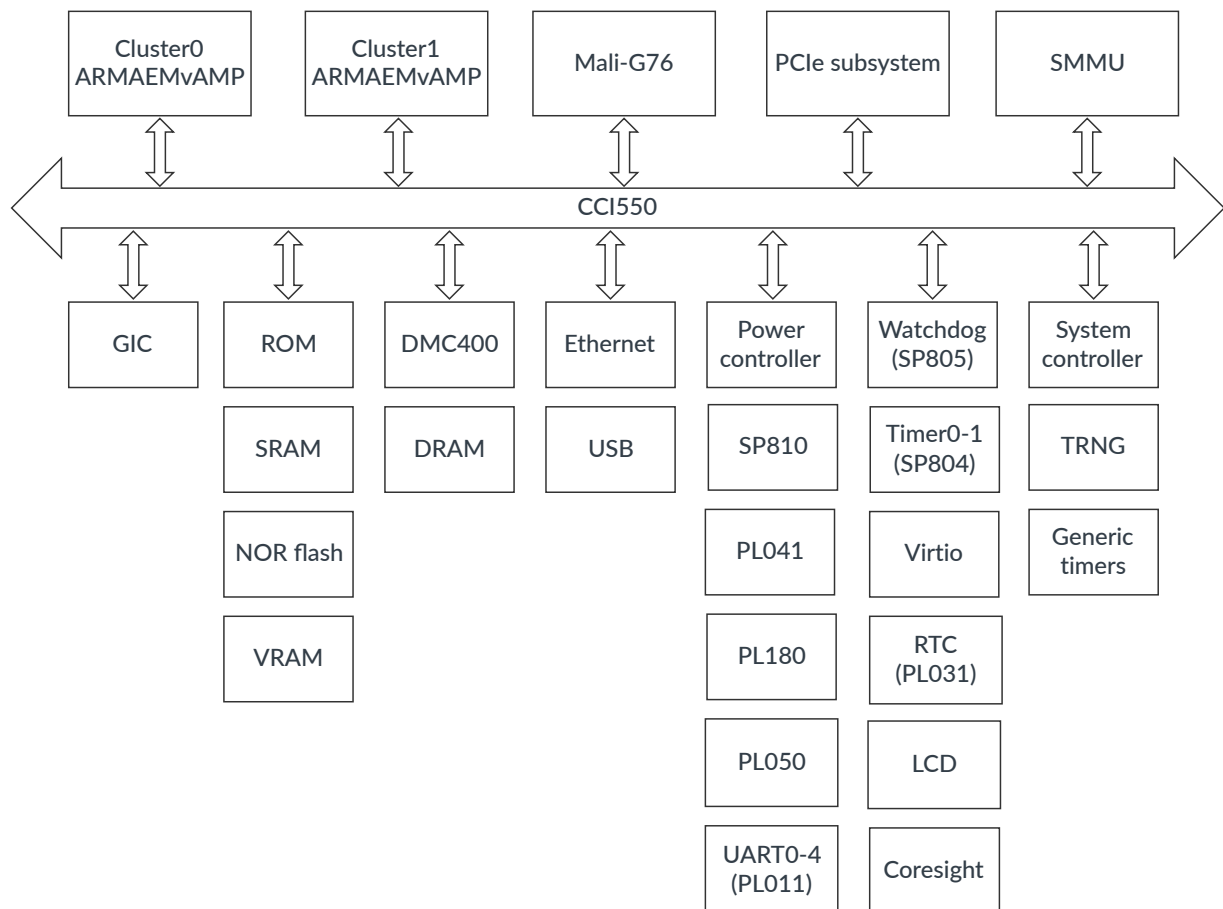
[Base Platform memory map](#) on page 1708

[Base - interrupt assignments](#) on page 1712

7.2.1 Base Platform RevC block diagram

This diagram shows the main components in the Base Platform RevC. The platform shown is FVP_Base_RevC-2xAEMvA.

Figure 7-1: Base Platform RevC block diagram



7.2.2 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+.



- 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 `pvbus2pci` 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.smmulogger
```

Use the [5.11 GenericTrace](#) on page 1646 plug-in to capture traces from the loggers. For example, you can see all accesses to device space by adding 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 [5.13 ListTraceSources](#) on page 1651 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 7-1: 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>pvbus_address_map_s</code>	PVBus	Slave	Input port to service transactions based on the PVBus protocol.
<code>pvbus_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>pvbus_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 <code>ClockSignal</code> provider.

Table 7-2: BasePlatformPCIRRevC 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 1581[VirtioBlockDevice](#) on page 1172

7.2.3 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_2xAEvMA.pci.pcie_rc.ROOT_COMPLEX_HIERARCHY"
```

- The top-level devices of the default tree are: [4.10.1 AHCI_SATA](#) on page 1191, [4.7.14 HostBridge](#) on page 1149, `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)

7.2.4 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_G_CBEID0` fields. The PMU is intended for demonstration purposes only and for driver development.

7.2.5 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
```

7.2.6 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:

```
BridgeInterrupt = (Device + DeviceInterrupt) % 4
```

This formula produces the following mappings:

BridgeInterrupt ID	DeviceInterrupt
200	INTA
201	INTB
202	INTC
203	INTD
207	SERR

7.2.7 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.

7.3 Base Platform startup configuration for v9 cores

The parameters `pctl.use_in_cluster_ppu` and `cluster.core_power_on_by_default` are used to configure startup of Arm®v9 cores in the Base Platform.

The following table shows the effect of each combination of these parameters:

Table 7-4: 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>	Only cores that are mentioned in the startup quad parameter are brought up. This configuration has the limitation that the RVBAR address must be supplied as a parameter on the command line. The application start address initialized by the application loader is overridden by the reset vector address provided in the parameter.	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

7.4 Base - memory

This section describes the memory of the Base Platform.

7.4.1 Base - secure memory

Enable security checking on memory transactions by the TZC-400 by using the `bp.secure_memory` parameter.

Table 7-5: 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.



The default state of the TZC-400 is to abort all accesses, even from Secure state.

Table 7-6: 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

7.4.2 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 [7.4.1 Base - secure memory](#) on page 1707.

Table 7-7: 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	0x00_0400_0000	512KB. This is the default size. To set SRAM to 256KB instead, set the parameter bp.secure_sram_size to 0.	0x00_0407_FFFF	S
Trusted DRAM	0x00_0600_0000	32MB	0x00_07FF_FFFF	S
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

Peripheral	Start address	Size	End address	Security
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
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

Peripheral	Start address	Size	End address	Security
GIC Physical CPU interface, GICC. You can configure the address of this region using model parameters. See 4.10.40 GICv3IRI on page 1425.	0x00_2C00_0000	8KB	0x00_2C00_1FFF	S/NS
GIC Virtual Interface Control, GICH. You can configure the address of this region using model parameters. See 4.10.40 GICv3IRI on page 1425.	0x00_2C01_0000	8KB	0x00_2C01_1FFF	S/NS
GIC Virtual CPU Interface, GICV. You can configure the address of this region using model parameters. See 4.10.40 GICv3IRI on page 1425.	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 AEMvA FVPs 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. You can configure the address of this region using model parameters. See 4.10.40 GICv3IRI on page 1425.	0x00_2F00_0000	64KB	0x00_2F00_FFFF	S/NS
GICv3 IRI GITS. You can configure the address of this region using model parameters. See 4.10.40 GICv3IRI on page 1425.	0x00_2F02_0000	128KB	0x00_2F03_FFFF	S/NS
GICv3 IRI GICR. You can configure the address of this region using model parameters. See 4.10.40 GICv3IRI on page 1425.	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

Peripheral	Start address	Size	End address	Security
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 1711

7.4.3 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 [0x0-0x7FFFFFFF] in the Base Platform are available at the same offset in the memory range [0x80000000-0xFFFFFFFF] in the BaseR Platform.
- Memory in the range [0x80000000-0xFFFFFFFF] in the Base Platform is available at the same offset in the range [0x0-0x7FFFFFFF] 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 0x0.

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 0x0. To move TCMs away from 0x0, use the parameters `itcm_base` and `dtcm_base`. For example:

```
-C cluster0.cpu0.itcm_base=0x8000 -C cluster0.cpu0.dtcm_base=0x18000
```

Related information

[Base Platform memory map](#) on page 1708

7.4.4 Base - 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: `0xCFDFDFDFDFDFDFCF`.



Memory is allocated on demand, and performance degrades if very large amounts of memory are used.

7.5 Base - 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 the *Arm® Generic Interrupt Controller Specification* describes.
- IRQ IDs 103-111, 192-194, and 200-207 apply to the Base Platform RevC only.

Table 7-8: 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.

IRQ ID	SPI offset	Device
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.
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.

IRQ ID	SPI offset	Device
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 7-9: PPI GIC assignments

IRQ ID	PPI offset	Device
19	3	Secure hypervisor virtual timer interrupt
20	4	Secure hypervisor physical timer interrupt
22	6	DCC, comms channel, interrupt
23	7	PMU, performance counter, overflow
24	8	CTI, Cross Trigger Interface, 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

7.6 Base - clocks

This section describes the clock frequencies of the Base Platform peripherals.

Table 7-10: 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
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

Device	Clock
Real-time Clock, PL031	1Hz

7.7 Base - parameters

This section describes the parameters.

Table 7-11: Base Platform parameters

Parameter	Type	Allowed values	Default value	Description
bp.dram_size	int	2, 4, or 8-4000000	4	Size of main memory in gigabytes: 2, 4, or any value between 8 and 4000000.
bp.proc_idn ³	uint32_t	-	_4	Processor ID for VE_SysRegs SYS_PROCID _n .
bp.secure_memory	bool	true, false	true	The security state of the processor limits access to peripherals and RAM. ⁵
bp.variant ³	uint32_t	0x0-0xF	_4	Board variant for VE_SysRegs SYS_ID.
cache_state_modelled	bool	true, false	true	Enable d-cache and i-cache state for all components.

Table 7-12: 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.

Related information

[VE_SysRegs - parameters](#) on page 1722

³ Some platforms do not expose this parameter.

⁴ Platform specific.

⁵ When loading an image on an EVS, you might see the following warning:

```
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, but this 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.

7.8 Base - components

This section describes the components.

7.8.1 Base - components - about

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.

7.8.2 Base - Base_PowerController component

This section describes the Base_PowerController component.

7.8.2.1 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 pct1.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.

7.8.2.2 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 7-13: 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 7-2: Power Control Processor Off Register bit assignments

[illegible]

Table 7-14: 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 7-3: Power Control Processor On Register bit assignments

[illegible]

Table 7-15: 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.

Configurations

Available in all configurations.

Attributes

See the register summary table.

Figure 7-4: Power Control Cluster Off Register bit assignments

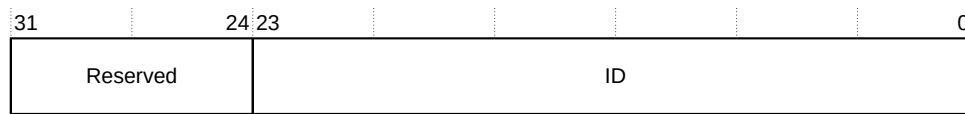


Table 7-16: Power Control Cluster Off Register bit assignments

Bits	Name	Function
[31:24]	-	Reserved.
[23:0]	ID	MPIDR format affinity value of powered-on processor in the cluster to be switched off. Programming error if MPIDR != self.

PWKUPR

The *Power Control Wakeup Register* (PWKUPR) characteristics are: purpose, usage constraints, configurations, and attributes.

Purpose

Configures whether wakeup requests from the GIC are enabled for this cluster.

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

See the register summary table.

Figure 7-5: Power Control Wakeup Register bit assignments

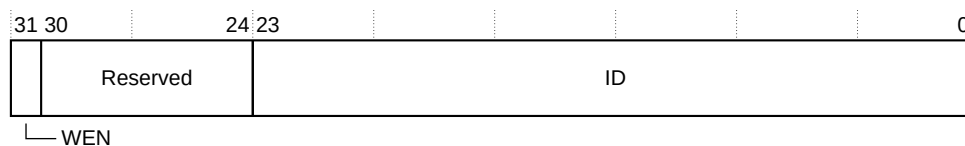


Table 7-17: Power Control Wakeup Register bit assignments

Bits	Name	Function
[31]	WEN	If set, enables wakeup interrupts (return from SUSPEND) for this cluster.
[30:24]	-	Reserved.
[23:0]	ID	MPIDR format affinity value of processor whose Wakeup Enable bit is to be configured.

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 7-6: Power Control SYS Status Register bit assignments

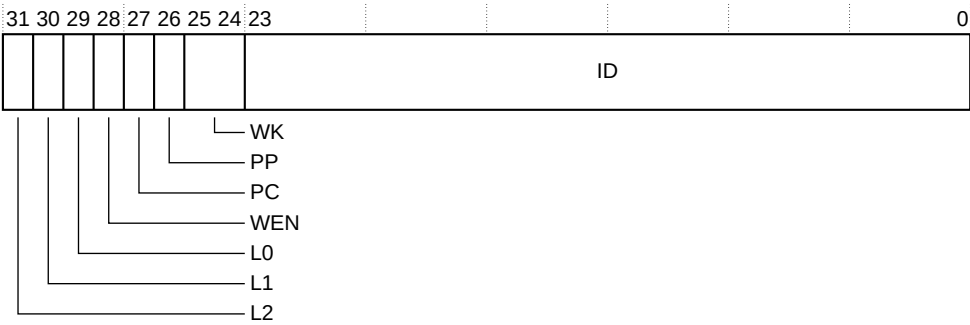


Table 7-18: Power Control SYS Status Register bit assignments

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 STANDBYWFI2.
[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.

Bits	Name	Function
[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.

7.8.3 Base - DebugAccessPort component

This section describes the DebugAccessPort component, a model of the *Debug Access Port* (DAP) for external debug connections.

7.8.3.1 DebugAccessPort - ports

This section describes the ports.

Table 7-19: DebugAccessPort ports

Name	Protocol	Type	Description
ap_pvbus_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.

7.8.3.2 DebugAccessPort - parameters

This section describes the parameters.

Table 7-20: Base Platform DebugAccessPort parameters

Name	Type	Allowed values	Default value	Description
ap0_rom_base_address	uint64_t	0x0-0xffffffffffffffff	0x0	ROM base address for AP0.
ap1_rom_base_address	uint64_t	0x0-0xffffffffffffffff	0x0	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.
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.

7.8.4 Simulator visualization - parameters

This section describes the parameters.

Table 7-21: 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. For example, 100 000 000 means 10 nanoseconds resolution simulated time for a 1Hz master clock. Used for recording. Higher values give a higher time resolution. Playback time base is always taken from the playback file.
recorder.verbose	-	0x0	Enables verbose messages. 0x0, no messages. 0x1, normal. 0x2, 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%.

7.8.5 Base - VE_SysRegs component

This section describes the VE system registers component.

7.8.5.1 VE_SysRegs - parameters

This section describes the parameters.

Table 7-22: Base Platform VE_SysRegs parameters

Name	Type	Allowed values	Default value	Description
exit_on_shutdown	bool	true, false	false	true: if software uses the SYS_CFGCTRL function SYS_CFG_SHUTDOWN, then the simulator shuts down and exits. ⁶
mmbSiteDefault	uint8_t	0x0-0x2	0x1	Default MMB source. 0x0, MB. 0x1, DB1. 0x2, DB2.
tilePresent	bool	true, false	true	Tile fitted.
user_switches_value	uint32_t	-	0x0	User switches.

Related information

[Base - parameters](#) on page 1715

7.8.5.2 VE_SysRegs - registers

This section describes the configuration registers.

Table 7-23: Base Platform 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.
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_PROCID2	0x8C	Read/write	Processor ID.

⁶ For more information on SYS_CFGCTRL, see the *Motherboard Express µATX V2M-P1 Technical Reference Manual*.

Name	Offset	Access	Description
SYS_PROCID3	0x90	Read/write	Processor ID.
SYS_CFGDATA	0xA0	Read/write	Data to read/write 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 controller.

7.9 Base - 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.

7.9.1 Base - VE compatibility - 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:

```
-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:

```
-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.

7.9.2 Base - VE compatibility - 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.*`

7.9.3 Base - VE compatibility - 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.

7.9.4 Base - VE compatibility - disable security

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 1708

7.10 Base - unsupported VE features

This system model does not support software that relies on some features of the VE model.

7.10.1 Base - unsupported VE features - 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.

7.10.2 Base - unsupported VE features - boot ROM alias at 0x00_0800_0000

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.

7.10.3 Base - unsupported VE features - change of older parameters

Most parameter names have been simplified between the VE model and the Base Platform system model.

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.

8. Microcontroller Prototyping System 2

This chapter describes the MPS2 system model.

8.1 MPS2 - about

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](#)

8.2 MPS2 platform types

Configure the MPS2 FVP platform type using the `fvmp_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: <ul style="list-style-type: none">• 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 . |

8.3 MPS2 - memory maps

This section describes the MPS2 memory maps.

8.3.1 MPS2 - memory map for models without the Arm®v8-M additions

This section describes the MPS2 memory map for older cores, without the Arm®v8-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 8-1: Memory map for models without the Arm®v8-M additions

Description	Modeled	Address range
Ethernet ⁷	Partial	0xA0000000-0xA000FFFF
PSRAM (16MB)	Yes	0x60000000-0x60FFFFFF
VGA Image (512x128) (AHB)	Yes	0x41100000-0x4110FFFF
VGA Console (AHB)	Yes	0x41000000-0x4100FFFF
Block RAM (boot time) ⁸	Yes	0x40200000-0x402FFFFFF
Reserved	N/A	0x40030000-0x401FFFFFF
SCC register	Yes	0x4002F000-0x4002FFFF
Reserved	N/A	0x40029000-0x4002EFFF
FPGA System Control & I/O, APB	Yes	0x40028000-0x40028FFF
Reserved	N/A	0x40025000-0x40027FFF
Audio I2S, APB	Partial	0x40024000-0x40024FFF
SBCon (Audio Configuration), APB	Yes	0x40023000-0x40023FFF
SBCon (Touch for LCD module), APB	Partial	0x40022000-0x40022FFF
PL022 (SPI for LCD module), APB	Partial	0x40021000-0x40021FFF
PL022 (SPI), APB	Yes	0x40020000-0x40020FFF
CMSDK system controller	Yes	0x4001F000-0x4001FFFF
Reserved for extra GPIO & other AHB peripherals	N/A	0x40012000-0x4001EFFF
CMSDK AHB GPIO #1	Yes	0x40011000-0x40011FFF
CMSDK AHB GPIO #0	Yes	0x40010000-0x40010FFF
CMSDK APB subsystem	Yes	0x40000000-0x4000FFFF

⁷ Through ahb_to_extmem16. Offset 0x0-0x0FE for CSRs, 0x100-0x1FE for FIFO.

⁸ Reserved 64KB, 16K implemented. This memory is wrapped through the region.

Description	Modeled	Address range
Reserved	N/A	0x20800000-0x20FFFFFF
ZBTSRAM 2 & 3 (2x32-bit) ⁹	Yes	0x20000000-0x207FFFFFF
Reserved	N/A	0x01010000-0x1FFFFFFF
Reserved	N/A	0x00800000-0x00FFFFFF
ZBTSRAM 1 (64-bit) ¹⁰	Yes	0x00400000-0x007FFFFFF
ZBTSRAM 1 (64-bit)	Yes	0x00004000-0x003FFFFFF
Mappable memory ¹¹	Yes	0x00000000-0x00003FFF

8.3.2 MPS2 - memory map for models with the Arm®v8-M additions

This section describes the MPS2 memory map for newer cores, with the Arm®v8-M additions.

For standard Arm® peripherals, see the TRM for that device.



Note

- 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 8-2: Memory map for models with the Arm®v8-M additions

Description	IDAU	Modeled	Address range
ZBTSRAM 1 (4MB) in <i>Non-secure</i> (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 0x007FFFFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI.	NS	N/A	0x00800000 to 0x0FFFFFFF
ZBTSRAM 1 (4MB) in <i>Secure</i> (S) world. Reserved 8MB, only 4MB implemented. Second half (4MB) aliased to first half (4MB).	S	Yes	0x10000000 to 0x107FFFFFF
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 0x207FFFFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI.	NS	N/A	0x20800000 to 0x20FFFFFF

⁹ Reserved 8MB, 4MB available. The two SRAM blocks are interleaved.

¹⁰ Wrapped. Only 4MB ZBTSRAM fitted.

¹¹ When `zbt_boot_ctrl` = 0, ZBTSRAM 1 is mapped to this region. Otherwise, `Remap_ctrl` = 0 maps Block RAM and `Remap_ctrl` = 1 maps ZBTSRAM 1. The V2M-MPS2 board microcontroller controls the `zbt_boot_ctrl` signal. The `zbt_boot_ctrl` signal overrides the boot option to enable use of the ZBT RAM.

Description	IDAU	Modeled	Address range
PSRAM (32MB)	NS	Yes	0x21000000 to 0x22FFFFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI.	NS	N/A	0x23000000 to 0x23FFFFFF
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. ¹²	NS	Yes	0x40000000 to 0x40000FFF
Timer 1. ¹²	NS	Yes	0x40001000 to 0x40001FFF
Dual Timer. ¹²	NS	Yes	0x40002000 to 0x40002FFF
Not used. ¹²	NS	N/A	0x40003000 to 0x40003FFF
UART #0. ¹²	NS	Yes	0x40004000 to 0x40004FFF
UART #1. ¹²	NS	Yes	0x40005000 to 0x40005FFF
UART #2. ¹²	NS	Yes	0x40006000 to 0x40006FFF
Not used. ¹²	NS	N/A	0x40007000 to 0x40007FFF

¹² Non-secure CMSDK APB subsystem.

Description	IDAU	Modeled	Address range
Watchdog. ¹²	NS	Yes	0x40008000 to 0x40008FFF
Not used. ¹²	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. ¹³	NS	N/A	0x40020000 to 0x40020FFF
PL022 (SPI for LCD). ¹³	NS	Partial	0x40021000 to 0x40021FFF
SBCon I2C (Touch for LCD). ¹³	NS	Partial	0x40022000 to 0x40022FFF
SBCon I2C (Audio configuration). ¹³	NS	Yes	0x40023000 to 0x40023FFF
Audio I2S. ¹³	NS	Partial	0x40024000 to 0x40024FFF

¹³ Non-secure FPGA APB subsystem (unused APB space: RAZ/WI).

Description	IDAU	Modeled	Address range
Not used. ¹³	NS	N/A	0x40025000 to 0x40027FFF
FPGA system control & I/O (LEDs, buttons...). ¹³	NS	Yes	0x40028000 to 0x40028FFF
Not used. ¹³	NS	N/A	0x40029000 to 0x4002EFFF
SCC registers. ¹³	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
Not used.	NS	N/A	0x40115000 to 0x401FFFFF
Ethernet (SMSC 91C111).	NS	Partial	0x40200000 to 0x402FFFFF
Not used. Default expansion port (MPS2 AHB subsystem): bus error or RAZ/WI.	NS	N/A	0x40300000 to 0x40FFFFFF
VGA console.	NS	Yes	0x41000000 to 0x4100FFFF
Not used.	NS	N/A	0x41010000 to 0x410FFFFF
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. ¹⁴	S	Yes	0x50000000 to 0x50000FFF
Timer 1. ¹⁴	S	Yes	0x50001000 to 0x50001FFF
Dual Timer. ¹⁴	S	Yes	0x50002000 to 0x50002FFF

¹⁴ Secure CMSDK APB subsystem.

Description	IDAU	Modeled	Address range
Not used. ¹⁴	S	N/A	0x50003000 to 0x50003FFF
UART #0. ¹⁴	S	Yes	0x50004000 to 0x50004FFF
UART #1. ¹⁴	S	Yes	0x50005000 to 0x50005FFF
UART #2. ¹⁴	S	Yes	0x50006000 to 0x50006FFF
Not used. ¹⁴	S	N/A	0x50007000 to 0x50007FFF
Watchdog. ¹⁴	S	Yes	0x50008000 to 0x50008FFF
Not used. ¹⁴	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
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

Description	IDAU	Modeled	Address range
FPGA APB subsystem:	S	-	0x50020000 to 0x5002FFFF
Not used. ¹⁵	S	N/A	0x50020000 to 0x50020FFF
PL022 (SPI for LCD). ¹⁵	S	Partial	0x50021000 to 0x50021FFF
SBCon I2C (touch for LCD). ¹⁵	S	Partial	0x50022000 to 0x50022FFF
SBCon I2C (audio configuration). ¹⁵	S	Yes	0x50023000 to 0x50023FFF
Audio I2S. ¹⁵	S	Partial	0x50024000 to 0x50024FFF
Not used. ¹⁵	S	N/A	0x50025000 to 0x50027FFF
FPGA system control & I/O (LEDs, buttons...). ¹⁵	S	Yes	0x50028000 to 0x50028FFF
Not used. ¹⁵	S	N/A	0x50029000 to 0x5002EFFF
SCC registers. ¹⁵	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

¹⁵ Secure FPGA APB subsystem.

Description	IDAU	Modeled	Address range
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. ¹⁶	S	Yes	0x58007000 to 0x58007FFF
Flash memory gating unit configuration (mapped to AHB port for CODE region in the bus matrix, not APB). ¹⁶	S	Yes	0x58008000 to 0x58009FFF
SRAM memory gating unit configuration (mapped to AHB port for SRAM region in the bus matrix, not APB). ¹⁶	S	Yes	0x5800A000 to 0x5800DFFF
Reserved. ¹⁶	S	N/A	0x5800E000 to 0x5800EFFF
Reserved. ¹⁶	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

¹⁶ Secure APB subsystem.

Description	IDAU	Modeled	Address range
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 0xF00FFFFF
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 0xFFFFFFFF

8.4 MPS2 - interrupt assignments

This section describes the interrupt assignments.

Table 8-3: 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.

Number	Interrupt
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. Arm®v8-M additions.
48	SPI #0. Arm®v8-M addition.
49	Reserved.
50	TRNG (Secure). Arm®v8-M addition.
51	Unique ID and Secure storage (Secure). Arm®v8-M addition.
52	DMA controller #0.
53	DMA controller #1.
54	SecureErrorIRQ. Arm®v8-M addition. ¹⁷

8.5 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

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.

¹⁷ 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.

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

The default value. 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.

9. Versatile Express Model

This chapter describes the components of Fast Models that are specific to the Versatile™ Express (VE) model of the hardware platform.

9.1 VE - about

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 and the instances in them, see [VE Platform FVPs](#) in the FVP Reference Guide.

LISA+ source and project files for the VE FVP examples 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.

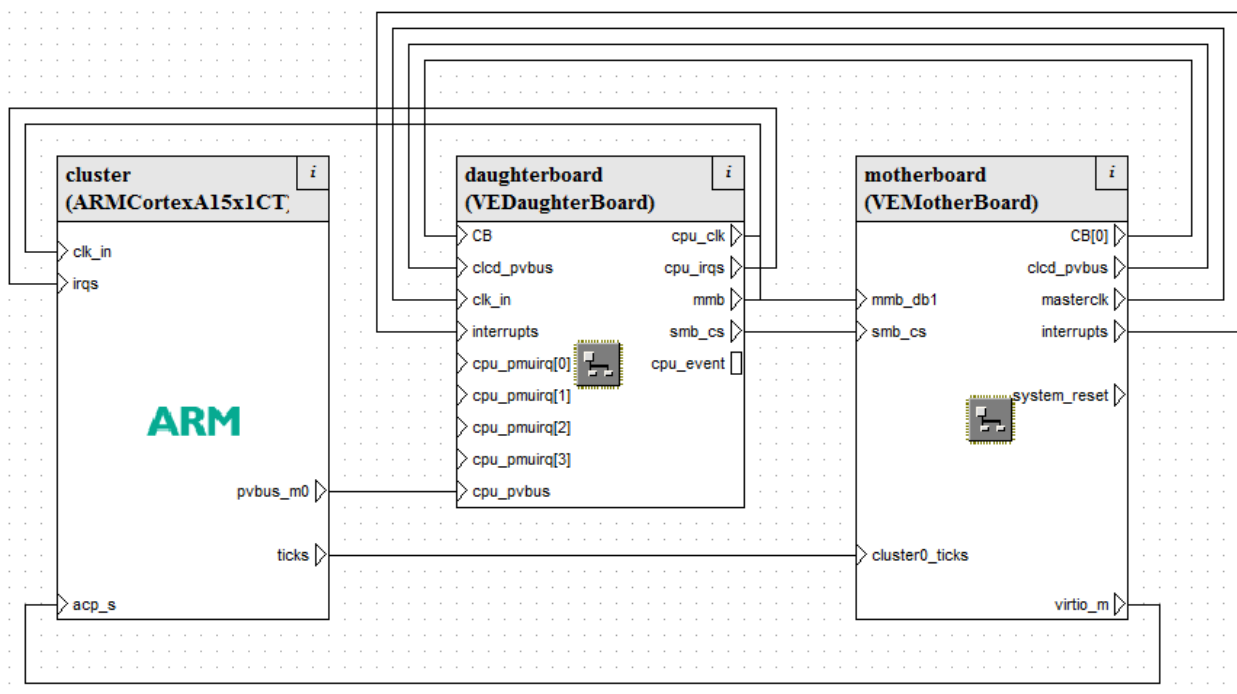


Arm bases the models on the VE platform memory map, but does not intend them to be accurate representations of a specific VE hardware revision. The VE FVPs support selected peripherals. The models are sufficiently complete and accurate to boot the same operating system images as the VE hardware.

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 9-1: Top-level block diagram of a VE model



Related information

[Getting Started with Fixed Virtual Platforms](#)

[Motherboard Express µATX V2M-P1 Technical Reference Manual](#)

9.2 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.



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 9-1: 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 CS2 region peripheral memory map , below.	Yes	0x00_18000000-0x00_1BFFFFFF	64MB
Peripherals (CS3). See CS3 region peripheral memory map , below.	Yes	0x00_1C000000-0x00_1FFFFFFF	64MB
CoreSight™ and peripherals	No	0x00_20000000-0x00_2CFFFFFF ¹⁸	-
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) ¹⁹	Yes	0x00_80000000-0x00_FFFFFFFF	2GB
Unused	-	0x01_00000000-0x07_FFFFFFFF	-
4GB DRAM (in 36-bit address space) ¹⁹	Yes	0x08_00000000-0x08_FFFFFFFF	4GB
Unused	-	0x09_00000000-0x7F_FFFFFFFF	-
4GB DRAM (in 40-bit address space) ¹⁹	Yes	0x80_00000000-0xFF_FFFFFFFF	4GB

The model has a `secure_memory` option. When you enable this option, the memory map changes for a number of peripherals.

Table 9-2: 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.

¹⁸ The private peripheral region address 0x2c000000 is mapped in this region. You can use the parameter `PERIPBASE` to map the peripherals to a different address.

¹⁹ The model contains a single 4GB block of DRAM, which is aliased across the three different 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 will be visible to reads at address 0x80_00000000. The lowest of the physical address regions is only 2GB in size.

Peripheral	Address range	Functionality with secure_memory enabled
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 9-3: CS2 region peripheral memory map

Peripheral	Modeled	Address range	Size	GIC Int ²⁰
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	-

Table 9-4: CS3 region peripheral memory map

Peripheral	Modeled	Address range	Size	GIC Int ²⁰
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_1C0FFFFF	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
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

²⁰ Use these interrupt signal values to program your interrupt controller. They are the SPI number plus 32. Add 32 to the interrupt numbers from the peripherals to form the interrupt number that the GIC sees. GIC interrupts 0-31 are for internal use.

Peripheral	Modeled	Address range	Size	GIC Int ²⁰
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	-

9.3 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 9-5: 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 ²¹	Yes	0xAE000000-0xAE000FFF
PL390 GIC Distributor ²¹	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
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

²¹ Cortex®-R4 and Cortex®-R5 models only.

Memory	Modeled	Address range
PL111 CLCD	Yes	0xB01F0000–0xB01FFFFF ²²
RAM	No	0xB4000000–0xBBFFFFFF
Video RAM	Yes	0xBC000000–0xBDFFFFFF
Ethernet (SMSC 91C111)	Yes	0xBE000000–0xBEFFFFFF
USB	No	0xBF000000–0xBFFFFFFF

9.4 VE - interrupt assignments for Cortex®-A series

The platform routes the following *Shared Peripheral Interrupts* (SPIs) to the GIC.

Table 9-6: 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
92	60	CPU 0 PMU
93	61	CPU 1 PMU
94	62	CPU 2 PMU
95	63	CPU 3 PMU
117	85	HDLCD

²² For Cortex®-R4 and Cortex®-R5 models, the range is 0xA0000000–0xA0010000.

9.5 VE - interrupt assignments for Cortex®-R series

This section describes the interrupt assignments.

Table 9-7: 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 9-8: 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
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

9.6 VE parameters

This section describes the VE FVP instantiation parameters.

9.6.1 VE instantiation parameters

This section describes the instantiation parameters for VE models.

Table 9-9: VE instantiation parameters

Component	Parameter	Type	Allowed values	Default value	Description
ve_sysregs	user_switches_value	Integer	See 9.6.3 VE switch S6 on page 1747.	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	-	0x12c	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.

9.6.2 VE secure memory parameters

This section describes the VE FVP secure memory parameters that you can change when you start the model.

Table 9-10: 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>

9.6.3 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 9-11: 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 9-12: 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.
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.

9.7 VE - components

A complete model implementation of the VE platform includes both VE-specific components and generic components, such as buses and timers.

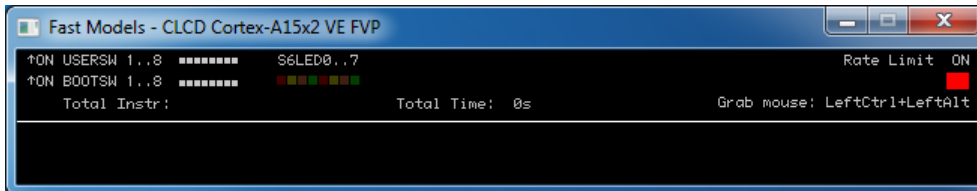
To see a list of all the component instances in the model, run it with the `--list-instances` option.

The generic components are documented in [4. Fast Models components](#) on page 105.

9.7.1 VEVisualisation component

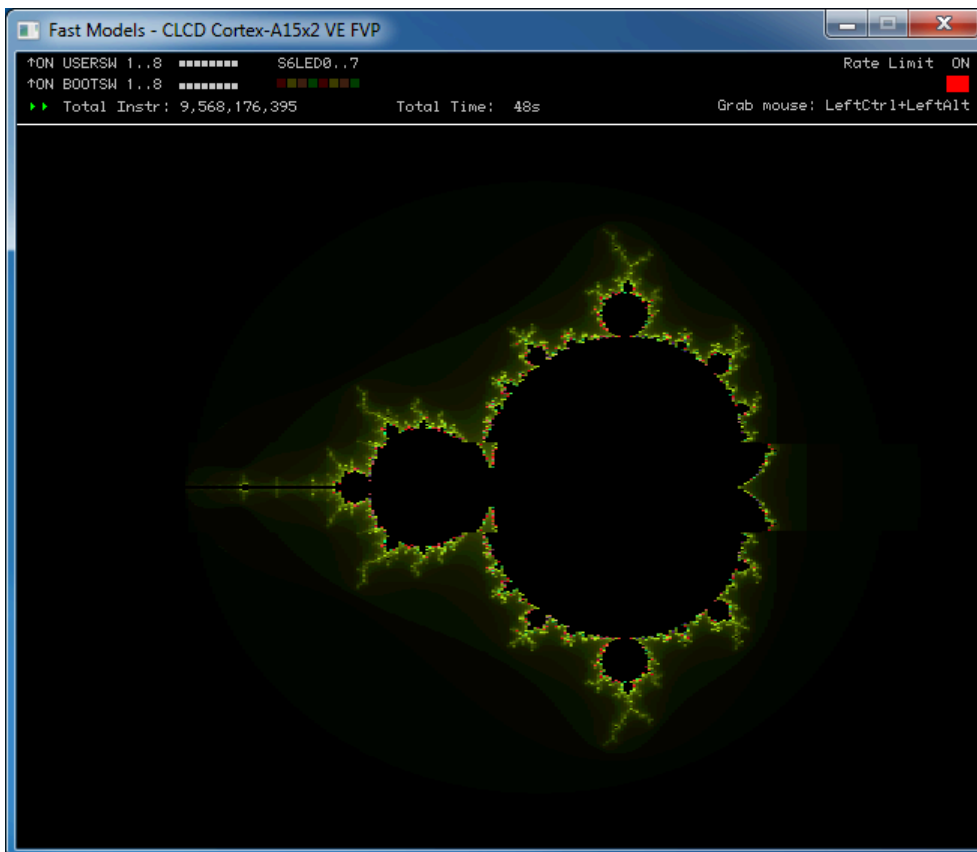
This component can generate events from the host mouse or keyboard when the visualization window is in focus. For example, you can toggle the switch elements from the visualization window.

Figure 9-2: 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 frame buffer.

Figure 9-3: VE FVP CLCD with brot.axf image



The VEVisualisation LISA+ component can be found in the `$PVLIB_HOME/examples/LISA/FVP_VE/LISA/` directory.



Using this component can reduce simulation performance. Use the `rate_limit_enable` parameter to control simulation speed.

Related information

[VEVisualisation - parameters](#) on page 1750

[Fixed Virtual Platforms FVP Reference Guide](#)

9.7.1.1 VEVisualisation - ports

This section describes the VEVisualisation component ports.

Table 9-13: VEVisualisation 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.

Related information

[KeyboardStatus protocol](#) on page 85

[LCD protocol](#) on page 85

[MouseStatus protocol](#) on page 87

9.7.1.2 VETVisualisation - 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
```



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, ensure `rate_limit-enable` is set to `false`, which is the default.

Table 9-14: VETVisualisation parameters

Name	Type	Allowed values	Default value	Description
<code>cluster0_name</code>	string	-	Cluster0	Label for cluster 0 performance values.
<code>cluster1_name</code>	string	-	Cluster1	Label for cluster 1 performance values.
<code>cluster2_name</code>	string	-	Cluster2	Label for cluster 2 performance values.
<code>cluster3_name</code>	string	-	Cluster3	Label for cluster 3 performance values.
<code>cpu_name</code>	string	-	-	Processor name displayed in window title.
<code>daughter_led_count</code>	int	0-32	0	Set to nonzero to display up to 32 LEDs. See the <code>daughter_leds</code> port.
<code>daughter_user_switch_count</code>	int	0-32	0	Set this parameter to display up to 32 switches. See the <code>daughter_user_switches</code> port.
<code>diagnostics</code>	bool	true, false	false	Print diagnostic messages.
<code>disable_visualisation</code>	bool	true, false	false	Disable the VETVisualisation component on model startup.
<code>rate_limit-enable</code>	bool	true, false	false	Restrict simulation speed so that simulation time more closely matches real time rather than running as fast as possible.
<code>recorder.checkInstructionCount</code>	bool	true, false	true	Check instruction count in recording file against actual instruction count during playback.
<code>recorder.playbackFileName</code>	string	-	""	Playback filename (empty string disables playback).
<code>recorder.recordingFileName</code>	string	-	""	Recording filename (empty string disables recording).
<code>recorder.recordingTimeBase</code>	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).
<code>recorder.verbose</code>	int	-	0	Enable verbose messages (1=normal, 2=even more).

Name	Type	Allowed values	Default value	Description
trap_key	int	Valid ATKeycode key value. See \$PVLIB_HOME/include/components/KeyCode.h 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. cpu_name replaces %cpu%.

9.7.1.3 VEEVisualisation - verification and testing

This component passes tests by use as an I/O device for booting Linux and other operating systems.

9.7.1.4 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.

9.7.1.5 VEEVisualisation - library dependencies

This component relies on the *Simple DirectMedia Layer* (SDL) libraries, specifically libSDL2-2.0.so.0.4.0.

This library is bundled with the Model Library 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 Cross-platform Development Library](#)

9.7.2 VE_SysRegs component

This section describes the VE system registers component.

9.7.2.1 VE_SysRegs - about

This LISA+ component is a model of the VE status and system control registers.

9.7.2.2 VE_SysRegs - ports

This section describes the ports.

Table 9-15: VE_SysRegs ports

Name	Protocol	Type	Description
cb[0-1]	VECBProtocol	Master	The <i>Configuration Bus</i> (CB) controls the power and reset sequence.
clock_24Mhz	ClockSignal	Slave	Reference clock for internal counter register.
clock_100Hz	ClockSignal	Slave	Reference clock for internal counter register.
clock_CLCD	ClockRateControl	Master	The clock for the LCD controller.
lcd	LCD	Master	Multimedia bus interface output to the LCD.
leds	ValueState	Master	Displays state of the SYS_LED register using the eight colored LEDs on the status bar.
mmb[0-2]	LCD	Slave	Multimedia bus interface input.
mmc_card_present	StateSignal	Slave	Indicates the presence of a <i>MultiMedia Card</i> (MMC) image.
pvbus	PVBus	Slave	Slave port for connection to PV bus master/decoder.
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.

9.7.2.3 VE_SysRegs - parameters

This section describes the parameters.

Table 9-16: VE_SysRegs parameters

Name	Type	Default value	Description
exit_on_shutdown	bool	false	Used to shut down the system. When <code>true</code> , if software uses the <code>SYS_CFGCTRL</code> function <code>SYS_CFG_SHUTDOWN</code> , then the simulator shuts down and exits. ²³
mmbSiteDefault	int	0x1	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.

²³ For more information on the `SYS_CFGCTRL` function values, see the *Motherboard Express µATX V2M-P1 Technical Reference Manual*.

9.7.2.4 VE_SysRegs - registers

This section describes the configuration registers.

Table 9-17: 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.
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.

9.7.2.5 VE_SysRegs - verification and testing

This component was tested as part of the Versatile™ Express model.

9.8 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 28

[Processor implementation](#) on page 33

9.8.1 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.

9.8.2 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`.

9.8.3 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 1741

9.8.4 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 PLO41 AACI PrimeCell and the audio CODEC as in the VE hardware, but with a limited number of sample rates.

9.8.5 Timing considerations for the VE FVPs

The Rate Limit feature matches simulation time to wall-clock time.

The *Fixed Virtual Platforms* (FVPs) provide an environment that enables running software applications 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.

When code interacts with real world devices like timers and keyboards, data arrives in the modeled device in real world, or wall clock, time, but simulation time can run much faster than the wall clock. This means that a single key press might register as several repeated key presses, or a single mouse click incorrectly becomes a double click.

Enabling Rate Limit, either using the Rate Limit button in the CLCD display, or the `rate_limit_enable` model instantiation parameter, forces the model to run at wall-clock time. This avoids issues with two clocks running at significantly different rates. Rate Limit is disabled by default. For interactive applications, Arm recommends enabling it.