Ncore 3.8 - Architecture Parameter Documentation

Rev: 0.39, September 23, 2025

ARTERIS® NCORE 3.8 - ARCHITECTURE PARAMETER DOCUMENTATION

Release Information

Version	Editor	Change	Date
0.39	BM	• Non-power of two interleaving updates • Updates the DCE interleaving parameter description • Adds a parameter to dmi for memory/interconnect view in the context of non-power of two interleaving • Updates the definition of the parameters for DMI interleaving to be closer to the actual object provided by Maestro to HW. Represnetation must change due to non-power of two interleaving. • Increase ncmdSkidBufSize for DCE.DMI and DII max is now 8448 entries. • Moved GIU interleaving function ot be next to the dmi ones.	09/18/2025
0.38	BM	• Removed nProcessors from CAIU parameter and replaced it with nExclusiveThreads (chapter 9.6) • Added derived parameter in DCE which were currently derived in tachl.cpr.(chapter 12) • Updated the name of GlobalLargestNProc to GlobalLargestNExclusiveThreads (chapter 3.11) • Fixed a typo in DII parameters where the name in the table and in the legend were different (nAddrTransRegsiters was called nLargestEndpint in the table) • nAddrTransRegisters maximum is increased to 16 for DMI.	08/22/2025
0.374	BM	• Update to nSttCtrlEntries for NCAIU • Adding a lot of missing non user visible system parameters (section 4.1)	08/07/25
0.373	BM	• NOttCtrlEntries aligned to Ncore 3.7 spec for CAIU and NCAIU	08/05/25
0.372	BM	• Update to CHI_B interface table tor restore fixed values • Adds GropExtId from 3.7 to CHI_E interface table • Adds default to CCP nWays parameter	08/04/2025
0.371	BM	• Update of nSttCtrlEntries for NCAIU per Bob's feedback	07/22/2025
0.37	BM	• Update to chapter 25 to clarify it only applies to single die (CONC-17490) • Adds wRequestorId to Concertocparams and requestermessage field (CONC-17453 1. And 2.) • Renames RemoteSnpCredits to nRemoteSnpCredits and updates the definition • Creation of nSttCtrlEntries for CAIU and NCAIU.	07/21/2025
0.36	BH	• Revised all single parameter tables with new table format • Added descriptions for the new table format in preface • Rename nUseMemRspIntrlv as enableReadRspIntrlv to align with Maestro • Effective revision and value of each parameter is now aligned with Ncore 3.7 • The status for 3.8 parameters are set to preview	07/09/2025
0.35	BM	• Update per Bernard Feedback • MessageSizeInGranule updated because data messages are larger than expected. • Update to wDid to remove the TargetId mention.	6/02/2025
0.34	*BM*	• Added msg_pri to ndp_order array in Table 19-2 :packet_descriptor~~Table 19-2 :packet_descriptor~~ • Update to CreditreturnBits to match maestro in Table 19-2 :packet_descriptor~~Table 19-2 :packet_descriptor~~ • Update MessageSizeInGranule to have 2 granule for VC1 (was way oversized) • New D2D system parameters related to DCE.	5/30/2025

Formatted: Font: (Default) Arial, 8 pt

Formatted: Font: (Default) Arial, 8 pt

		• Added a description in QosEnable parameter • Added AddresBits parameter to DMI which was missing • Removed Cmd credit from NCAIU which were deprecated in 3.4 • Removed Snoop filter assignment from inside NCAIU which I believe is not correct. • Add a cxs_async_adapter unit (22.5) • Made min/max columns of derived/fix socket parameter N/A if fixed. • widthAdapter and rateAdapter update to have 512 has a valid width • add a table of tables • finished reformatting.	
0.33	BM	• MAES-8165 • Adds RemoteSnpCredit • Reformating of the document • Updated a typo in nAiuConnectedDces • Removed credit parameters in CAIU deprecated in 3.4. • Removed mrd credit parameters in DCE which were deprecated in 3.4. • Removed smoop filter parameter from CAIU which I don't think exists. • Replaced Aprimarybit by its actual name : SetSelectPrimaryBitV • Changed hasSystemCache in dmi to useCache • Added useAtomic parameter which was missing in DMI	5/27/2025
0.32	BM	• New number of GIUs parameter in Assumptions section (**Error! Reference source not found.**) • New non-user visible system section 4. • Added the GIU interleaving parameters to section 4 • Added a parameter which contains the number of dies (section 3.11) • Remove CXSCNTL from the table, it is not a property of the interface already has the with in wCntl and this table is not a list of signals. Removed CXS_ASYNC from the table. This is not a property of the CXS socket. A CXS async adapter will be inserted if the clock of the socket and of the GIU are different similarly to CHI.	3/30/2025

0.31	HL	• Added to following signals the CXS.B interface in ~~Table 23-8 CXS Parameters~~Table 23-8 CXS Parameters
		  ○ CXS_ASYNC
		  ○ CXSCNTL
		  ○ CXSMAXPKTPERFLIT
		• Re-based to the latest version of 3.7, which is version 0.81 and reviewed the previous version:
		  ○ Removed the redundant tables already exist in single Ncore/chiplet configuration in section
			1/9/2025

0.30	BM	• Addition of concerto parameter table for multi die: sections 25.1 and 25.2 • Addition of GIU user settable parameter in section 0 • Addition of CXS socket: section 23.8 • Addition of CXS user settable parameter: section 7.10	12/23/2024

Legend:

BM	Benjamin Madon
HL	Hao Luan
BH	Brian Huang
Xx	Whoever else edited this document

Active	3.2	User-GUI

Note:

Formatted: Font: 8 pt

Issues to be discussed:

Confidential Proprietary Notice

Confidentiality Status

This document is Confidential and Proprietary. This document may only be used and distributed in accordance with the terms of the agreement entered into by Arteris IP and the party that Arteris IP delivered this document to.

Product Status

The information in this document is ***Preliminary***.

Web Address

http://www.arteris.com

Field Code Changed

Table of Contents

PREFACE ..... 19~~26~~

1. OVERVIEW ..... 22~~29~~

2. ASSUMPTIONS ..... 23~~30~~

2.1. System Constraints ..... 23~~30~~

3. SYSTEM USER SETTABLE PARAMETERS ..... 25~~32~~

3.1. Project name and RTL prefix parameters ..... 25~~32~~
3.2. System level connectivity parameters ..... 26~~33~~
3.3. System address map parameters ..... 29~~36~~
3.4. System resiliency parameters ..... 31~~38~~
3.5. System error parameters ..... 34~~41~~
3.6. System QoS parameters ..... 35~~42~~
3.7. System level debug parameters ..... 37~~44~~
3.8. System level physical parameters ..... 39~~46~~
3.9. System engineering parameters ..... 40~~47~~
3.10. RISC-V related parameters ..... 41~~48~~
3.11. D2D system parameters ..... 41~~48~~

4. SYSTEM NON-USER VISIBLE / DERIVED PARAMETERS ..... 43~~50~~

4.1. System parameters ..... 43~~50~~
4.2. GIU interleaving parameters ..... 47~~54~~

5. POWER AND CLOCK USER SETTABLE PARAMETERS ..... 50~~57~~

6. MEMORY MAP USER SETTABLE PARAMETERS ..... 53~~60~~

7. SOCKET USER SETTABLE PARAMETERS ..... 62~~69~~

7.1. Native interface parameters ..... 62~~69~~
7.1.1. Smallest Coherent Configurations for Ncore ..... 62~~69~~
7.2. AXI4 Interface ..... 63~~70~~
7.3. AXI5 Interface ..... 66~~73~~
7.4. ACE Interface ..... 70~~77~~
7.5. ACE5 Interface ..... 75~~82~~
7.6. ACE5-LITE Interface ..... 80~~87~~
7.7. ACE-LITE Interface ..... 87~~94~~
7.8. CHI Issue B Interface ..... 90~~97~~
7.9. CHI Issue E Interface ..... 94~~101~~
7.10. CXS.B Interface ..... 97~~104~~

8. CONCERTO USER SETTABLE PARAMETERS ..... 98~~105~~

9. CAIU USER SETTABLE PARAMETERS ..... 99~~106~~

9.1. CAIU resource parameters ..... 99~~106~~
9.2. CAIU credit parameters ..... 101~~108~~
9.3. CAIU address map parameter ..... 102~~109~~
9.4. CAIU snoop filter parameters ..... 104~~111~~

9.5. CAIU PERFORMANCE COUNTER PARAMETERS 104111
9.6. CAIU PROCESSOR INFO PARAMETERS 105112
9.7. CAIU SYSCMD HARDWARE PARAMETERS 106113
9.8. CAIU CONNECTIVITY PARAMETERS 108115
10. CAIU NON-USER SETTABLE PARAMETERS 111118
11. DCE USER SETTABLE PARAMETERS 112119
11.1. DCE RESOURCE PARAMETERS 112119
11.2. DCE CREDIT PARAMETERS 113120
11.3. DCE SNOOP FILTER PARAMETERS 114121
11.4. DCE PERFORMANCE COUNTER PARAMETERS 117124
11.5. DCE EXCLUSIVE MONITOR PARAMETERS 117124
11.6. DCE CONNECTIVITY PARAMETERS 118125
12. DCE NON-USER SETTABLE PARAMETERS 122129
13. DMI USER SETTABLE PARAMETERS 123130
13.1. DMI RESOURCE PARAMETERS 123130
13.2. DMI ADDRESS MAP PARAMETERS 127134
13.3. DMI SYSTEM CACHE PARAMETERS 127134
13.4. DMI PERFORMANCE COUNTER PARAMETERS 131138
13.5. DMI ATOMIC PARAMETERS 131138
13.6. DMI TRANSACTION TABLE RESERVATION FOR IMPROVED QOS 133140
13.7. DMI ADDRESSBITS PARAMETER FOR CUSTOM AXIID ADDRESS BIT SELECT 135142
14. DII USER SETTABLE PARAMETERS 136143
14.1. DII RESOURCE PARAMETERS 136143
14.2. DII ADDRESS MAP PARAMETERS 138145
14.3. DII PERFORMANCE MONITOR PARAMETERS 138145
14.4. DII ADDRESSBITS PARAMETER FOR CUSTOM AXIID ADDRESS BIT SELECT 140147
15. DVE USER SETTABLE PARAMETERS 142149
15.1. DVE RESOURCE PARAMETERS 142149
15.2. DVE PERFORMANCE MONITOR PARAMETERS 142149
15.3. DVM SYSTEM LEVEL PARAMETERS 144151
16. NCAIU USER SETTABLE PARAMETERS 146153
16.1. NCAIU MULTIPORT PARAMETERS 146153
16.2. NCAIU RESOURCE PARAMETERS 147154
16.3. NCAIU CREDIT PARAMETERS 147154
16.4. NCAIU ADDRESS MAP PARAMETER 148155
16.5. NCAIU SNOOP FILTER PARAMETERS 149156
16.6. NCAIU PROXY CACHE PARAMETERS 149156
16.7. NCAIU PERFORMANCE COUNTER PARAMETERS 151158
16.8. NCAIU DISABLE READ DATA INTERLEAVING PARAMETERS 151158
16.9. NCAIU SYSCMD HARDWARE PARAMETERS 153160
16.10. NCAIU CONNECTIVITY PARAMETERS 154161
16.11. NCAIU OWO PARAMETERS 156163
17. NCAIU NON-USER SETTABLE PARAMETERS 158165
18. GIU USER SETTABLE PARAMETERS 158165

18.1. VIRTUAL CHANNEL CREDITS 158165
19. GIU NON-USER SETTABLE PARAMETER 159166
20. CACHE AND SNOOP FILTER USER SETTABLE PARAMETERS 161168
21. TILING RELATED PARAMETERS 164171
22. LEGATO USER SETTABLE PARAMETERS 165172
22.1. SYM_SWITCH/SYM_BUF_SWITCH 165172
22.2. SYM_ASYNC_ADAPTER 166173
22.3. CHI_ASYNC_ADAPTER 166173
22.4. SYM_NRATE_ADAPTER 166173
22.5. CXS_ASYNC_ADAPTER 166173
22.6. DW_ADAPTER 167174
22.7. SYM_PIPE_ADAPTER 167174
23. DERIVED/FIXED SOCKET PARAMETERS 168175
23.1. AXI INTERFACE 168175
23.2. APB INTERFACE 168175
23.3. ACE INTERFACE 169176
23.4. ACE5-LITE INTERFACE 169176
23.5. ACE-LITE INTERFACE 170177
23.6. CHI_B INTERFACE 170177
23.7. CHI_E INTERFACE 172179
23.8. CXS.B INTERFACE 174181
24. SINGLE DIE VARIANT DERIVED/FIXED CONCERTO PARAMETERS 175182
24.1. SYSTEM PARAMETER : CONCERTOCSMIPARAMS 175182
24.2. SYSTEM PARAMETER : CONCERTOCPARAMS 178185
24.3. SYSTEM PARAMETER : CONCERTOCREQUESTMESSAGEFIELDS 181188
24.4. SYSTEM PARAMETER : CONCERTOCRESPONSEMESSAGEFIELDS 185192
25. MULTI-DIE VARIANT OF DERIVED/FIXED CONCERTO PARAMETERS 189196
25.1. SYSTEM PARAMETER : CONCERTOCSMIPARAMS (MULTI-DIE) 189196
25.2. SYSTEM PARAMETER : CONCERTOCPARAMS 191198
25.3. SYSTEM PARAMETER : CONCERTOCREQUESTMESSAGEFIELDS 193200
25.4. SYSTEM PARAMETER : CONCERTOCRESPONSEMESSAGEFIELDS 197204
26. LEGATO DERIVED/FIXED PARAMETERS 202209
26.1. PMA 202209
26.2. SYM_ASYNC_ADAPTER 203210
26.3. SYM_BUF_SWITCH 204211
26.3.1. Configuration details 204211
26.4. SYM_IBUF_SWITCH 206213
26.5. WIDTH/RATE_ADAPTER (SYM_NRATE_ADAPTER) 207214
26.6. SYM_PIPE_ADPATER 211218
26.7. INTERRUPT 211218
26.8. PARAMETER FOR CSR NETWORK 211218
26.9. CHI_ASYNC_ADAPTER 211218
26.10. CXS_ASYNC_ADAPTER 211218
26.11. CSR FIXED PARAMETERS 212219

26.11.1. *Atut_apb parameters* 212219
26.11.2. *Atui_axi parameters* 214221
26.11.3. *APB socket parameters* 215222
26.11.4. *AXI socket parameters* 216223
26.11.5. *Switch parameters* 217224

27. OTHER USER SETTABLE PARAMETERS 218225

27.1. PARAMETER RELATED WITH PLACEHOLDER GENERIC SIGNAL 218225
27.2. PARAMETER RELATED WITH SRAM ASSIGNMENT 220227
27.2.1. *SW_memory* 220227
27.2.2. *Generic ports* 222229

28. USER SETTABLE PARAMETER FOR SYNTHESIS 224231

Table of Figures

FIGURE 3-1 NAIUPORTS PARAMETER 2835
FIGURE 5-1: POWER AND CLOCK DOMAIN DEFINITION FOR NCORE 3.6 5057
FIGURE 2 PROPOSED ARCHITECTURE TO TAKE ADVANTAGE OF TT RESERVATIONS 133140
FIGURE 22-1: SYM_BUF_SWITCH IN CDTI, WITH ONLY ONE VC 166173
FIGURE 26-1: PMA IN CLOCK DOMAIN 202209
FIGURE 26-2: SYM_BUF_SWITCH IN CDTI, WITH ONLY ONE VC 204211
FIGURE 26-3: SYM_BUF_SWITCH IN CDTI 206213

TABLE 2-1: NUMBER OF COHERENT-AGENT INTERFACE UNITS(CAIU) 23
TABLE 2-2: NUMBER OF NON-COHERENT AGENT INTERFACE UNITS(NC-AIU) 23
TABLE 2-3: NUMBER OF DISTRIBUTED MEMORY INTERFACES 24
TABLE 2-4: NUMBER OF SNOOP FILTERS 24
TABLE 2-5: NUMBER OF DISTRIBUTED VIRTUAL MEMORY SYSTEM ENGINES 24
TABLE 2-6: NUMBER OF DISTRIBUTED COHERENCY ENGINES 24
TABLE 2-7: NUMBER OF DISTRIBUTED IO INTERFACES 25
TABLE 2-8: NUMBER OF GIU 25
TABLE 3-1: PROJECTNAME PARAMETER 25
TABLE 3-2: USERTLPREFIX PARAMETER 26
TABLE 3-3: COHERENTTEMPLATE PARAMETER 26
TABLE 3-4: NAIUPORTS PARAMETER 27
TABLE 3-5: APRIMARYAIUPORTBITS PARAMETER 27
TABLE 3-6: ASECONDARYAIUPORTBITS PARAMETER 27
TABLE 3-7: NGPRA PARAMETER 29
TABLE 3-8: DCEINTERLEAVINGPRIMARYBITS 29
TABLE 3-9: DCEINTERLEAVINGSECONDARYBITS 29
TABLE 3-10: RESILIENCE ON/OFF PARAMETER 31
TABLE 3-11: SAFETYCONFIG PARAMETER 31
TABLE 3-12: DUPLICATION ENABLE PARAMETER 31
TABLE 3-13: NATIVE INTERFACE PROTECTION PARAMETER 32
TABLE 3-14: INTERUNITDELAY PARAMETER 32
TABLE 3-15: RESILIENCYPROTECTIONTYPE PARAMETER 32
TABLE 3-16: MEMORYPROTECTIONTYPE PARAMETER 33
TABLE 3-17: FNDISABLERESILIENCYBISTDEBUGPIN PARAMETER 33
TABLE 3-18: TIMEOUTTHRESHOLDPARAMETER 34
TABLE 3-19: MEMORYPROTECTIONTYPE PARAMETER 34
TABLE 3-20: QOSENABLED PARAMETER 35
TABLE 3-21: QOSMAP PARAMETER 35
TABLE 3-22: QOSEVENTTHRESHOLD PARAMETER 36
TABLE 3-23: NMAINTRACEBUFSIZE PARAMETER 37
TABLE 3-24: NTRACEREGISTERS PARAMETER 37
TABLE 3-25: NUNITTRACEBUFSIZES PARAMETER 37
TABLE 3-26: FNDEBUGAPBENABLE PARAMETER 38
TABLE 3-27: SYNCDEPTH PARAMETER 39
TABLE 3-28: ASSERTIONENABLE PARAMETER 40
TABLE 3-29: ENGVERID PARAMETER 40
TABLE 3-30: IMPLVERID PARAMETER 40
TABLE 3-31: NODVM PARAMETER 41
TABLE 3-32: NDIES PARAMETERS 41
TABLE 3-33: NREMOTESNPCREDITS PARAMETERS 41
TABLE 3-34: GLOBALLARGESTNEXCLUSIVETHREADS PARAMETERS 42
TABLE 3-35: NGLOBALCACHINGAGENTS PARAMETERS 42
TABLE 4-1 NDCES PARAMETER 43
TABLE 4-2 NDIIS PARAMETER 43
TABLE 4-3 NDMIS PARAMETER 43
TABLE 4-4 NAIUS PARAMETER 44
TABLE 4-5 NNCAIUS PARAMETER 44
TABLE 4-6 NCAIUS PARAMETER 44
TABLE 4-7 NDVES PARAMETER 44
TABLE 4-8 NPROXYCACHES PARAMETER 45

TABLE 4-9 NCHIAAIUS PARAMETER 45
TABLE 4-10 NCHIBAIUS PARAMETER 45
TABLE 4-11 NACEAIUS PARAMETER 45
TABLE 4-12 NACELITEAIUS PARAMETER 46
TABLE 4-13 NACELITEEAIUS PARAMETER 46
TABLE 4-14 NAXIAIUS PARAMETER 46
TABLE 4-15 WFUNITID PARAMETER 47
TABLE 4-16 WNUNITID PARAMETER 47
TABLE 4-17: REMOTEINTERLEAVINGOBJECT OBJECT REPRESENTING THE INTERLEAVING OF GIUS 47
TABLE 5-1: PARAMETER RELATED WITH **CLOCK REGION**: FREQUENCY 51
TABLE 5-2: PARAMETER RELATED WITH **CLOCK REGION**: UNITCLOCKGATING 51
TABLE 5-3: PARAMETER RELATED WITH **CLOCK DOMAIN**: GATING 52
TABLE 6-1: PARAMETER RELATED WITH **CSR REGION**: MEMORYBASE 53
TABLE 6-2: PARAMETER RELATED WITH **CSR REGION**: MEMORYSIZE 53
TABLE 6-3: PARAMETER RELATED WITH **BOOT REGION**: MEMORYBASE 54
TABLE 6-4: PARAMETER RELATED WITH **BOOT REGION**: MEMORYSIZE 54
TABLE 6-5: PARAMETER RELATED WITH **BOOT REGION**: MG_REF 54
TABLE 6-6: PARAMETER RELATED WITH **BOOT REGION**: CHANNEL_REF 54
TABLE 6-7: PRIMARYINERLEAVINGBITONE 55
TABLE 6-8: PRIMARYINERLEAVINGBITTWO 55
TABLE 6-9: PRIMARYINERLEAVINGBITTHREE 55
TABLE 6-10: PRIMARYINERLEAVINGBITFOUR 56
TABLE 6-10: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.TWOWAYINTRLVFUNCV 56
TABLE 6-10: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.THREEWAYINTRLVFUNCV 57
TABLE 6-10: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.FOURWAYINTRLVFUNCV 57
TABLE 6-10: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.SIXWAYINTRLVFUNCV 58
TABLE 6-10: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.EIGHTWAYINTRLVFUNCV 58
TABLE 6-10: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.TWELVEWAYINTRLVFUNCV 59
TABLE 6-10: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.SIXTEENWAYINTRLVFUNCV 59
TABLE 4-18: GIU2WIFV GIU 2 WAY INTERLEAVING FUNCTION 60
TABLE 4-19: GIU3WIFV GIU 3 WAY INTERLEAVING FUNCTION 60
TABLE 4-18: GIU2WIFV GIU 4 WAY INTERLEAVING FUNCTION 60
TABLE 7-1: FNNATIVEINTERFACE PARAMETER 62
TABLE 7-2: PARAMETER RELATED WITH AXI4 INTERFACE: WARID 63
TABLE 7-3: PARAMETER RELATED WITH AXI4 INTERFACE: WAWID 63
TABLE 7-4: PARAMETER RELATED WITH AXI4 INTERFACE: WADDR 63
TABLE 7-5: PARAMETER RELATED WITH AXI4 INTERFACE: WDATA 64
TABLE 7-6: PARAMETER RELATED WITH AXI4 INTERFACE: AWUSER 64
TABLE 7-7: PARAMETER RELATED WITH AXI4 INTERFACE: ARUSER 64
TABLE 7-8: EXCLUSIVE_ACCESSES PROPERTY FOR AXI4 INTERFACE 65
TABLE 7-9: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR AXI4 INTERFACE 65
TABLE 7-10: PARAMETER RELATED WITH AXI5 INTERFACE: WARID 66
TABLE 7-11: PARAMETER RELATED WITH AXI5 INTERFACE: WAWID 66
TABLE 7-12: PARAMETER RELATED WITH AXI5 INTERFACE: WADDR 66
TABLE 7-13: PARAMETER RELATED WITH AXI5 INTERFACE: WDATA 67
TABLE 7-14: PARAMETER RELATED WITH AXI5 INTERFACE: AWUSER 67
TABLE 7-15: PARAMETER RELATED WITH AXI5 INTERFACE: ARUSER 67
TABLE 7-16: EXCLUSIVE_ACCESSES PROPERTY FOR AXI5 INTERFACE 67
TABLE 7-17: CHECK_TYPE PROPERTY FOR AXI5 INTERFACE 68
TABLE 7-18: ATOMIC_TRANSACTIONS PROPERTY FOR AXI5 INTERFACE 68
TABLE 7-19: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR AXI5 INTERFACE 68

TABLE 7-20: PARAMETER RELATED WITH ACE INTERFACE: WARID 70
TABLE 7-21: PARAMETER RELATED WITH ACE INTERFACE: WAWID 70
TABLE 7-22: PARAMETER RELATED WITH ACE INTERFACE: WADDR 70
TABLE 7-23: PARAMETER RELATED WITH ACE INTERFACE: WDATA 71
TABLE 7-24: PARAMETER RELATED WITH ACE INTERFACE: AWUSER 71
TABLE 7-25: PARAMETER RELATED WITH ACE INTERFACE: ARUSER 71
TABLE 7-26: EXCLUSIVE_ACCESSES PROPERTY FOR ACE INTERFACE 71
TABLE 7-27: PARAMETER RELATED TO EVENTIN INTERFACE 72
TABLE 7-28: PARAMETER RELATED TO EVENTOUT INTERFACE 72
TABLE 7-29: SHAREABLE_TRANSACTIONS PROPERTY FOR ACE INTERFACE 72
TABLE 7-30: CONTINUOUS_CACHE_LINE_READ_DATA PROPERTY FOR ACE INTERFACE 72
TABLE 7-31: DVM_V8 PROPERTY FOR ACE INTERFACE 73
TABLE 7-32: DVM_V8.1 PROPERTY FOR ACE INTERFACE 73
TABLE 7-33: DVM_MESSAGE_SUPPORT PROPERTY FOR ACE INTERFACE 73
TABLE 7-34: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR ACE INTERFACE 74
TABLE 7-35: PARAMETER RELATED WITH ACE5 INTERFACE: WARID 75
TABLE 7-36: PARAMETER RELATED WITH ACE5 INTERFACE: WAWID 75
TABLE 7-37: PARAMETER RELATED WITH ACE5 INTERFACE: WADDR 75
TABLE 7-38: PARAMETER RELATED WITH ACE5 INTERFACE: WDATA 75
TABLE 7-39: PARAMETER RELATED WITH ACE5 INTERFACE: AWUSER 76
TABLE 7-40: PARAMETER RELATED WITH ACE5 INTERFACE: ARUSER 76
TABLE 7-41: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR ACE5 INTERFACE 76
TABLE 7-42: PARAMETER RELATED TO EVENTIN INTERFACE 77
TABLE 7-43: PARAMETER RELATED TO EVENTOUT INTERFACE 77
TABLE 7-44: CHECK_TYPE PROPERTY FOR ACE5 INTERFACE 77
TABLE 7-45: EXCLUSIVE_ACCESSES PROPERTY FOR ACE5 INTERFACE 77
TABLE 7-46: SHAREABLE_TRANSACTIONS PROPERTY FOR ACE5 INTERFACE 78
TABLE 7-47: CONTINUOUS_CACHE_LINE_READ_DATA PROPERTY FOR ACE5 INTERFACE 78
TABLE 7-48: DVM_V8 PROPERTY FOR ACE5 INTERFACE 78
TABLE 7-49: DVM_V8.1 PROPERTY FOR ACE5 INTERFACE 79
TABLE 7-50: DVM_MESSAGE_SUPPORT PROPERTY FOR ACE5 INTERFACE 79
TABLE 7-51: PARAMETER RELATED WITH ACE5-LITE/ACE5-LITEDVM INTERFACE: WARID 80
TABLE 7-52: PARAMETER RELATED WITH ACE5-LITE/ACE5-LITEDVM INTERFACE: WAWID 80
TABLE 7-53: PARAMETER RELATED WITH ACE5-LITE/ACE5-LITEDVM INTERFACE: WADDR 80
TABLE 7-54: PARAMETER RELATED WITH ACE5-LITE/ACE5-LITEDVM INTERFACE: WDATA 80
TABLE 7-55: PARAMETER RELATED WITH ACE5-LITE/ACE5-LITEDVM INTERFACE: AWUSER 82
TABLE 7-56: PARAMETER RELATED WITH ACE5-LITE/ACE5-LITEDVM INTERFACE: ARUSER 82
TABLE 7-57: PARAMETER TO ENABLE DVM FUNCTIONALITIES FOR THE ACE5-LITE INTERFACE: DVM 82
TABLE 7-58: DVM_V8 PROPERTY FOR ACE5-LITEDVM INTERFACE 83
TABLE 7-59: DVM_V8.1 PROPERTY FOR ACE5-LITEDVM INTERFACE 83
TABLE 7-60: DVM_V8.4 PROPERTY FOR ACE5-LITEDVM INTERFACE 83
TABLE 7-61: DVM_MESSAGE_SUPPORT PROPERTY FOR ACE5-LITEDVM INTERFACE 83
TABLE 7-62: PARAMETER RELATED TO EVENTOUT INTERFACE 84
TABLE 7-63: PARAMETER TO ENABLE THE PROTECTION ON THE ACE5-LITE/ACE5-LITEDVM INTERFACE 84
TABLE 7-64: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR ACE5-LITEDVM INTERFACE 84
TABLE 7-65: ATOMIC_TRANSACTIONS PROPERTY FOR ACE5-LITE/ACE5-LITEDVM INTERFACE 85
TABLE 7-66: EXCLUSIVE_ACCESSES PROPERTY FOR ACE5-LITE/ACE5-LITEDVM INTERFACE 85
TABLE 7-67: SHAREABLE_TRANSACTIONS PROPERTY FOR ACE5-LITE/ACE5-LITEDVM INTERFACE 86
TABLE 7-68: CACHE_STASH_TRANSACTIONS PROPERTY FOR ACE5-LITE INTERFACE 86
TABLE 7-69: TRACE_SIGNALS PROPERTY FOR ACE5-LITE INTERFACE 86
TABLE 7-70: PARAMETER RELATED WITH ACE-LITE INTERFACE: WARID 87

TABLE 7-71: PARAMETER RELATED WITH ACE-LITE INTERFACE: WAWID 87
TABLE 7-72: PARAMETER RELATED WITH ACE-LITE INTERFACE: WADDR 87
TABLE 7-73: PARAMETER RELATED WITH ACE-LITE INTERFACE: WDATA 87
TABLE 7-74: PARAMETER RELATED WITH ACE-LITE INTERFACE: AWUSER 88
TABLE 7-75: PARAMETER RELATED WITH ACE-LITE INTERFACE: ARUSER 88
TABLE 7-76: PARAMETER RELATED WITH ACE-LITE INTERFACE: EAC 88
TABLE 7-77: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR ACE-LITE INTERFACE 89
TABLE 7-78: SHAREABLE_TRANSACTIONS PROPERTY FOR ACE-LITE INTERFACE 89
TABLE 7-79: EXCLUSIVE_ACCESSES PROPERTY FOR ACE-LITE INTERFACE 89
TABLE 7-80: PARAMETER RELATED WITH CHI_B_INTERFACE: NODEID_WIDTH 90
TABLE 7-81: PARAMETER RELATED WITH CHI_B_INTERFACE: WADDR 90
TABLE 7-82: PARAMETER RELATED WITH CHI_B_INTERFACE: REQ_RSVDC 90
TABLE 7-83: PARAMETER RELATED WITH CHI_B_INTERFACE: WDATA 90
TABLE 7-84: PARAMETER RELATED WITH CHI_B_INTERFACE: ENPOISON 92
TABLE 7-85: PARAMETER RELATED TO SYSCO INTERFACE 92
TABLE 7-86: PARAMETER RELATED TO EVENTIN INTERFACE 92
TABLE 7-87: PARAMETER RELATED TO EVENTOUT INTERFACE 93
TABLE 7-88: PARAMETER TO ENABLE THE PROTECTION ON THE CHI-B INTERFACE 93
TABLE 7-89: PARAMETER RELATED WITH CHI_E_INTERFACE: NODEID_WIDTH 94
TABLE 7-90: PARAMETER RELATED WITH CHI_E_INTERFACE: WADDR 94
TABLE 7-91: PARAMETER RELATED WITH CHI_E_INTERFACE: REQ_RSVDC 94
TABLE 7-92: PARAMETER RELATED WITH CHI_E_INTERFACE: WDATA 94
TABLE 7-93: PARAMETER RELATED WITH CHI_E_INTERFACE: ENPOISON 95
TABLE 7-94: PARAMETER RELATED TO SYSCO INTERFACE 95
TABLE 7-95: PARAMETER RELATED TO EVENTIN INTERFACE 95
TABLE 7-96: PARAMETER RELATED TO EVENTOUT INTERFACE 96
TABLE 7-97: PARAMETER TO ENABLE THE PROTECTION ON THE CHI-E INTERFACE 96
TABLE 7-98: PARAMETER TO SELECT THE NUMBER OF CXS.B LINK CREDITS 97
TABLE 7-99: PARAMETER TO SELECT CXS.B CHECK_TYPE 97
TABLE 9-1: NOTTCTRLENTRIES FOR CAIU 99
TABLE 9-2: GENERIC PORTS PARAMETER FOR CAIU 99
TABLE 9-3: MEMORY PARAMETER FOR CAIU 99
TABLE 9-4: NNATIVECREDITS PARAMETER FOR CAIU 101
TABLE 9-5: NSTASHSNPCREDITS FOR CAIU 101
TABLE 9-6: NDCECMDCREDITS FOR CAIU 101
TABLE 9-7: NDMICMDCREDITS FOR CAIU 102
TABLE 9-8: NDIICMDCREDITS FOR CAIU 102
TABLE 9-9: FNCSRACCESS_PARAMETER 102
TABLE 9-10: SNOOPFILTER_REF PARAMETER FOR CAIUS 104
TABLE 9-11: CAIU PERFORMANCE COUNTER PARAMETERS 104
TABLE 9-12: CAIU LATENCY COUNTER PARAMETERS 104
TABLE 9-13: NPROCESSOR PARAMETERS FOR CAIU 105
TABLE 9-14: NEXCLUSIVETHREADS PARAMETERS FOR CAIU 106
TABLE 9-15: AXIDPROCSELECTBITS PARAMETERS FOR CAIU 106
TABLE 9-16: USESYSCOENGINE PARAMETERS FOR CAIU 106
TABLE 9-17: USESYSREQSENDER PARAMETERS FOR CAIU 108
TABLE 9-18: USESYSREQRECEIVER PARAMETERS FOR CAIU 108
TABLE 9-19: HEXAIUDCEVEC PARAMETERS FOR CAIU 108
TABLE 9-20: HEXAIUDMIVEC PARAMETERS FOR CAIU 109
TABLE 9-21: HEXAIUDIIVEC PARAMETERS FOR CAIU 109
TABLE 9-22: HEXAIUCONNECTEDDCEFUNITID PARAMETERS FOR CAIU 109

TABLE 9-23: NAIUCONNECTEDDCES PARAMETERS FOR CAIU 110
TABLE 10-1: NSTTCTRLENTRIES FOR CAIU 111
TABLE 11-1: NATTCTRLENTRIES PARAMETER 112
TABLE 11-2: NCMDSKIDBUFSIZE PARAMETER 112
TABLE 11-3: NCMDSKIDBUFARB PARAMETER 113
TABLE 11-4: NDCERBCREDITS PARAMETER 113
TABLE 11-5: NAIUSNPCREDITS PARAMETER 113
TABLE 11-6: NDMIMRDCREDITS PARAMETER 114
TABLE 11-7: NSETS SF CONFIGURATION PARAMETERS 114
TABLE 11-8: NWAYS SF CONFIGURATION PARAMETERS 114
TABLE 11-9: NVICTIMENTRIES SF CONFIGURATION PARAMETERS 115
TABLE 11-10: SETSELECTPRIMARYBITV PARAMETERS 115
TABLE 11-11: MEMORY PARAMETER FOR DCE SNOOP FILTER 115
TABLE 11-12: REMOTECACHINGAGENTS PARAMETER 116
TABLE 11-13: DCE NPERFCOUNTERS PARAMETERS 117
TABLE 11-14: NTAGGEDMONITORS PARAMETER 117
TABLE 11-15: HEXDCECONNECTEDDMIFUNITID PARAMETER 118
TABLE 11-16: HEXDCECONNECTEDCAFUNITID PARAMETER 118
TABLE 11-17: HEXDCEDMIVEC PARAMETER 118
TABLE 11-18: HEXDCEDMIRBOFFSET PARAMETER 120
TABLE 11-19: NDCECONNECTEDCAS PARAMETER 120
TABLE 11-20: NDCECONNECTEDDMIS PARAMETER 120
TABLE 11-21: NDCERBCREDITS PARAMETER 121
TABLE 12-1: NLPIDPERCAAGENT FOR DCE 122
TABLE 12-2: NEXCLUSIVETHREAD FOR DCE 122
TABLE 13-1: GENERICPORTS PARAMETERS FOR DMI 123
TABLE 13-2: NRTTCTRLENTRY PARAMETERS 123
TABLE 13-3: NWTTCTRLENTRY PARAMETERS 123
TABLE 13-4: NDMIRBCREDITS PARAMETER 123
TABLE 13-5: NCMDSKIDBUFSIZE PARAMETER 124
TABLE 13-6: NCMDSKIDBUFARB PARAMETER 124
TABLE 13-7: NMRDSKIDBUFSIZE PARAMETER 125
TABLE 13-8: NMRDSKIDBUFARB PARAMETER 125
TABLE 13-9: ENABLEREADRSPINTRLV PARAMETER 126
TABLE 13-10: NEXCLUSIVEENTRIES PARAMETER 126
TABLE 13-11: NADDRTRANSREGISTERS PARAMETERS 127
TABLE 13-12: DIRECTMEMORYMODE PARAMETERS 127
TABLE 13-13: DMI SYSTEM CACHE ENABLE PARAMETERS 127
TABLE 13-14: DMI SCRATCHPAD ENABLE PARAMETERS 128
TABLE 13-15: DMI CACHE WAY PARTITIONING REGISTERS PARAMETERS 129
TABLE 13-16: DMI CACHE NTAGBANK CONFIGURATION PARAMETERS 129
TABLE 13-17: DMI CACHE NDATABANK CONFIGURATION PARAMETERS 129
TABLE 13-18: MEMORY PARAMETER SMC 129
TABLE 13-19: DMI NPERFCOUNTERS PARAMETERS 131
TABLE 13-20: DMI LATENCY COUNTER PARAMETERS 131
TABLE 13-21: USEATOMIC PARAMETERS 131
TABLE 13-22: DMIQOSTHVAL PARAMETERS 133
TABLE 13-23: NDMIWTTQOSRSV PARAMETERS 134
TABLE 13-24: NDMIRTTQOSRSV PARAMETERS 134
TABLE 13-25: ADDRESSBITS PARAMETER 135
TABLE 14-1: NRTTCTRLENTRY PARAMETERS 136

TABLE 14-2: NWTTCTRLENTRY PARAMETERS 136
TABLE 14-3: NLARGESTENDPOINT PARAMETER 136
TABLE 14-4: NDIIRBCREDITS PARAMETER 137
TABLE 14-5: NCMDSKIDBUFSIZE PARAMETER 137
TABLE 14-6: NCMDSKIDBUFARB PARAMETER 137
TABLE 14-7: NEXCLUSIVEENTRIES PARAMETER 138
TABLE 14-8: NADDRTRANSREGISTERS PARAMETER FOR DII 138
TABLE 14-9: DII NPERFCOUNTERS PARAMETERS 138
TABLE 14-10: DII LATENCY COUNTER PARAMETERS 140
TABLE 14-11: ADDRESSBITS PARAMETER 140
TABLE 15-1: MEMORY PARAMETER FOR DVE 142
TABLE 15-2: DVE EVENTBROADCASTERFIFODEPTH PARAMETER 142
TABLE 15-3: DVE NPERFCOUNTERS PARAMETERS 142
TABLE 15-4: DVMVERSIONSUPPORT PARAMETER 144
TABLE 15-5: NDVMCMDCREDITS PARAMETER 144
TABLE 15-6: NDVMSNPCREDITS PARAMETER 144
TABLE 16-1: NNATIVEINTERFACEPORTS PARAMETER FOR NCAIU 146
TABLE 16-2: ANCAIUINTVFUNC PARAMETER FOR NCAIU 146
TABLE 16-3: NOTTCTRLENTRIES FOR NCAIU 147
TABLE 16-4: MEMORY PARAMETER FOR NCAIU 147
TABLE 16-5: NDCECMDCREDITS FOR NCAIU 147
TABLE 16-6: NDMICMDCREDITS FOR NCAIU 148
TABLE 16-7: NDIICMDCREDITS FOR NCAIU 148
TABLE 16-8: FNCSRACCESS_PARAMETER 148
TABLE 16-9: SNOOPFILTERASSIGNMENT PARAMETER 149
TABLE 16-10: PROXY CACHE ENABLE PARAMETERS 149
TABLE 16-11: PROXY CACHE NTAGBANK CONFIGURATION PARAMETERS 149
TABLE 16-12: PROXY CACHE NDATABANK CONFIGURATION PARAMETERS 150
TABLE 16-13: PROXY CACHE NSETS CONFIGURATION PARAMETERS 150
TABLE 16-14: PROXY CACHE NWAYS CONFIGURATION PARAMETERS 150
TABLE 16-15: NPERFCOUNTERS PARMAETER FOR NCAIU 151
TABLE 16-16: NLATENCYCOUNTERS PARMAETER FOR NCAIU 151
TABLE 16-17: FNDISABLERDINTERLEAVE PARMAETER FOR NCAIU 151
TABLE 16-18: USESYSCOENGINE PARAMETERS FOR NCAIU 153
TABLE 16-19: USESYSREQSENDER PARAMETERS FOR NCAIU 153
TABLE 16-20: USESYSREQRECEIVER PARAMETERS FOR NCAIU 153
TABLE 16-21: HEXAIUDCEVEC PARAMETERS FOR NCAIU 154
TABLE 16-22: HEXAIUDMIVEC PARAMETERS FOR NCAIU 154
TABLE 16-23: HEXAIUDIIVEC PARAMETERS FOR NCAIU 154
TABLE 16-24: HEXAIUCONNECTEDDCEFUNITID PARAMETERS FOR NCAIU 156
TABLE 16-25: NAIUCONNECTEDDCES PARAMETERS FOR NCAIU 156
TABLE 16-26: ORDEREDWRITEOBSERVATION PARAMETER 156
TABLE 16-27: MULTICYCLEODSRAM PARAMETER 156
TABLE 16-28: IOAIUPDATAWIDTH PARAMETER 157
TABLE 16-29: NENTRIESINSHIM PARAMETER 157
TABLE 17-1: NSTTCTRLENTRIES FOR NCAIU 158
TABLE 18-1 : VC_DESCRIPTOR.NUMBEROFCREDIT PARAMETER 158
TABLE 19-1: VC_DESCRIPTOR PARAMETER 159
TABLE 19-2 :PACKET_DESCRIPTOR PARAMETER 159
TABLE 20-1: NSETS PARAMETERS OF CCP 161
TABLE 20-2: NWAYS PARAMETER OF CCP 161

TABLE 20-3: USESCRATCHPAD PARAMETER OF CCP 161
TABLE 20-4: PRISUBDIAGADDRBITS PARAMETERS 162
TABLE 20-5: TAGBANKSELBITS PARAMETERS 162
TABLE 20-6: DATABANKSELBITS PARAMETERS 162
TABLE 20-7: CACHEREPLPOLICY PARAMETER 162
TABLE 20-8: CACHEREPLSTATEWIDTH PARAMETER 163
TABLE 21-1: PORT_ID PARAMETER FOR PACKETIZER 164
TABLE 21-2: ROUTEONSID PARAMETER FOR PACKETIZER 164
TABLE 22-1: SYM_BUF_SWITCH AND SYM_SWITCH PARAMETER: PORTDATAWIDTH 165
TABLE 22-2: SYM_BUF_SWITCH PARAMETER: INPUTBUFFERDEPTH 165
TABLE 22-3: SYM_PIPE_ADAPTER PARAMETER: DEPTH 167
TABLE 23-1: AXI INTERFACE FIXED PARAMETERS 168
TABLE 23-2: APB INTERFACE FIXED PARAMETERS 168
TABLE 23-3: ACE INTERFACE USER SETTABLE PARAMETERS 169
TABLE 23-4: ACE5-LITE INTERFACE DERIVED PARAMETERS 169
TABLE 23-5: ACE-LITE INTERFACE DERIVED PARAMETERS 170
TABLE 23-6: CHI_B INTERFACE DERIVED PARAMETERS-1 170
TABLE 23-7: CHI_E INTERFACE DERIVED PARAMETERS-1 172
TABLE 23-8 CXS PARAMETERS 174
TABLE 24-1: CONCERTOCSMIPARAMS PARAMETERS 175
TABLE 24-2: CONCERTOCPARAMS PARAMETERS 178
TABLE 24-3: CONCERTOCREQUESTMESSAGEFIELDS PARAMETERS 181
TABLE 24-4: CONCERTOCRESPONSEMESSAGEFIELDS PARAMETERS 185
TABLE 25-1: CONCERTOCSMIPARAM PARAMETERS FOR MULTI-DIE CONFIGURATION 189
TABLE 25-2 CONCERTOCPARAMS PARAMETERS FOR MULTI-DIE VARIANT 191
TABLE 25-3: CONCERTOCREQUESTMESSAGEFIELDS PARAMETERS 193
TABLE 25-4: CONCERTOCRESPONSEMESSAGEFIELDS PARAMETERS 197
TABLE 26-1: SYM_ASYNC_ADAPTER 203
TABLE 26-2: SYM_BUF_SWITCH 204
TABLE 26-3: SYM_IBUF_SWITCH 206
TABLE 26-4: NINPUTWIDTH PARAMETERS FOR WIDTHADAPTER 207
TABLE 26-5: NOUTPUTWIDTH PARAMETERS FOR WIDTHADAPTER 207
TABLE 26-6: BOOLPIPELINE PARAMETERS FOR WIDTHADAPTER 208
TABLE 26-7: NDEPTH PARAMETERS FOR WIDTHADAPTER 208
TABLE 26-8: NWIDTH PARAMETERS FOR RATEADAPTER 209
TABLE 26-9: NDEPTH PARAMETERS FOR WIDTHADAPTER 209
TABLE 26-10: INSERTION RULES 210
TABLE 26-11: SYM_PIPE_ADAPTER 211
TABLE 26-12: ATUT_APB BLOCK PARAMETERS 212
TABLE 26-13: ATUI_AXI BLOCK PARAMETERS 214
TABLE 26-14: APB SOCKET PARAMETERS 215
TABLE 26-15: AXI SOCKET PARAMETERS 216
TABLE 26-16: SWITCH BLOCK PARAMETERS 217
TABLE 27-1: PARAMETER FOR GENERIC PORT: WIRENAME 218
TABLE 27-2: PARAMETER FOR GENERIC PORT: WIREWIDTH 218
TABLE 27-3: PARAMETER FOR GENERIC PORT: WIRERTLPREFIX 218
TABLE 27-4: PARAMETER FOR GENERIC PORT: DIRECTION 218
TABLE 27-5: MEMORY PARAMETER FOR IOAIU 220
TABLE 27-6: MEMORY PARAMETER FOR SNOOP FILTER 220
TABLE 27-7: MEMORY PARAMETER FOR DCE 220
TABLE 27-8: MEMORY PARAMETER FOR DMI 220

TABLE 27-9: MEMORY PARAMETER FOR CCP 221
TABLE 27-10: MEMORY PARAMETER FOR DVE 221
TABLE 27-11: MEMORY PARAMETER FOR DII 221
TABLE 27-12: ENHALFSPEEDDATASRAM PARAMETER 221
TABLE 27-13: ENSRAMPIPE PARAMETER 221
TABLE 27-14: PARAMETER FOR SRAM GENERIC PORT: WIRENAME 222
TABLE 27-15: PARAMETER FOR SRAM GENERIC PORT: WIREWIDTH 222
TABLE 27-16: PARAMETER FOR SRAM GENERIC PORT: DIRECTION 222
TABLE 28-1: SYNTHESIS PARAMETER: CHECKONLY 224
TABLE 28-2: SYNTHESIS PARAMETER: TOPOMODE 224
TABLE 28-3: SYNTHESIS PARAMETER: TECHNOLOGY 224
TABLE 28-4: SYNTHESIS PARAMETER: CLOCKUNCERTAINTY 225
TABLE 28-5: SYNTHESIS PARAMETER: RTLWRAPPERDIR 225
TABLE 28-6: SYNTHESIS PARAMETER: HARDMACRODBS 225
TABLE 28-7: SYNTHESIS PARAMETER: BOTTOMUPSYNTHESIS 225
TABLE 28-8: SYNTHESIS PARAMETER: MAXTRANSITION 226
TABLE 28-9: SYNTHESIS PARAMETER: OUTPUT LOAD 227
TABLE 28-10: SYNTHESIS PARAMETER: ULVTPERCENTAGE 227
TABLE 28-11: SYNTHESIS PARAMETER: COMPILECOMMAND 227

Preface

This preface introduces the Arteris® Network-on-Chip Hierarchical Coherency Engine Architecture Specification.

About this document

This technical document is for the Arteris Network-on-Chip Hierarchical Coherency Engine Architecture. It describes the subsystems and their function along with the system's interactions with the external subsystems. It also provides reference documentation and contains programming details for registers.

Parameter Table Format

The following describes the format used by the parameter table:

Parameter name – describe the name of the parameter

Value – describe the data type, and configurable options of corresponding parameter

Constraint/Dependency – describe the limitation/dependency related to other parameters or use cases.

Customer description – functional description of the parameter

Engineering description – Additional detail functional description in implementation and limitations.

Release Info – parameter status and alignment with Ncore release version

Status – Can be active, deprecated, preview or experimental

Effective version – The starting version for the parameter to be effective

Visibility :

– Engineering : visible by engineering team, not visible to customers

– User-GUI: User visible and can be configured by Maestro

– User-Register: User visible and is configured by corresponding programming register

Change History – Describe the value/limitation/use cases/status changes in different release

Product revision status

TBD

Intended audience

This manual is for system designers, system integrators, and programmers who are designing or programming a System-on-Chip (SoC) that uses or intend to use the Arteris Network-on-Chip Hierarchical Coherency System (ANoC-HCS).

Using this document

TBD

Glossary

The Arteris© Glossary is a list of terms used in Arteris© documentation, together with definitions for those terms. The Arteris© Glossary does not contain terms that are industry standard unless the Arteris© meaning differs from the generally accepted meaning.

Typographic conventions

italic

Introduces special terminology, denotes cross-references, and citations.

bold

Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.

monospace

Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.

monospace italic

Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name. monospace italic Denotes arguments to monospace text where the argument is to be replaced by a specific value. monospace bold Denotes language keywords when used outside example code.

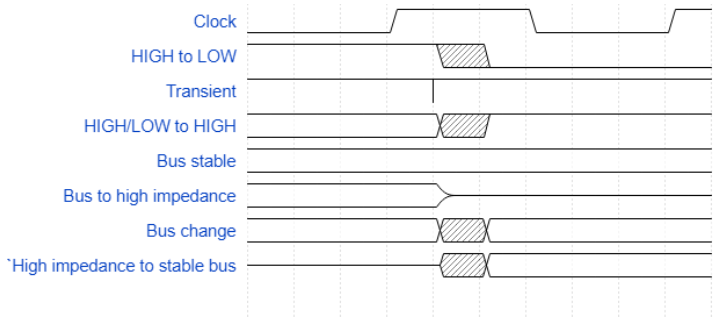
SMALL CAPITALS

Used in body text for a few terms that have specific technical meanings, that are defined in the Arteris® Glossary. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

Timing diagrams

The following figure explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.



Signals

The signal conventions are:

Signal level

The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW.
Asserted means:

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

Lowercase n

At the start or end of a signal name denotes an active-LOW signal.

Definitions

Flow

Communication between two end points in the protocol. Includes sending a message from the initiator of a transaction (sender), for example an AIU, to the completer of the transaction (receiver), and returning a response back.

Target

The endpoint of a flow, Ncore architecture implements the following targets:

- DCE, DCE - commands from CAIU, NCAIU
- DMI - commands from CAIU, NCAIU and DCE; data from CAIU, NCAIU
- DII - commands & data from CAIU, NCAIU
- CAIU - snoop commands from DCE

1. *Overview*

This document describes parameters which are related with RTL and DV implementation. Parameters that are software centric will be described in another document.The main purpose of this document is to enumerate parameter for Ncore 3.8. Therefore, several parameters for future purpose could be omitted.

2. *Assumptions*

NCore 3.x has three parameter categories:

- pre-map parameters,
- post-map parameters which are being defined in Maestro, and

  hw-cpr files which are being defined in CPR file, mainly by HW design team.

Premap parameters are being used at Maestro mapping stage, and then post-map parameters override the values after the mapping. Finally, hw-cpr files will be used on top of results of software, and main purpose of this file is to define derivation rules from pre-map/post-map software-type files. This document is only summarizing pre-map and post-map parameters of Maestro.

However, this document does not specify pre-map/post-map parameters. Instead, divide the parameters into (1) user settable parameters and (2) derived/fixed parameters. User parameter part will be visible to the customer through tcl configuration and GUI configuration. The default, min, and max value of the user settable parameters must match with user settable ranges.

2.1. System Constraints

System constraints do **NOT** correspond any user settable parameter. It is to add checks so that user cannot add more components over the limits.

TABLE 2-1: NUMBER OF COHERENT-AGENT INTERFACE UNITS(CAIU)

Parameter Name	**Number of CAIUs**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
			Min: 1 Max: 32	N/A
Constraint/Dependency	Constrained by total throughput provided the total number of DCEs			
Customer Description	Number of CAIUs			
Engineering Description	The total number of Coherent-Agent Interface Units Configured in an Ncore Interconnect			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 2-2: NUMBER OF NON-COHERENT AGENT INTERFACE UNITS(NC-AIU)

Parameter Name	**Number of NCAIUs**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
			Min: 0 Max: 32	N/A
Constraint/Dependency				
Customer Description	Number of NCAIUs			
Engineering Description	The total number of Non-coherent-Agent Interface Units configured in an Ncore Interconnect			

Release Info	Status	Effective version	Visibility
	Active	3.2	Engineering
Change History			

Field Code Changed

TABLE 2-3: NUMBER OF DISTRIBUTED MEMORY INTERFACES

Parameter Name	Number of DMIs			
Value	Data Type	Architecture	Release	Default
			Min: 1 Max: 16	N/A
Constraint/Dependency	Mainly constrained by the total throughput provided by the number of DCEs			
Customer Description	Number of DMIs			
Engineering Description	Total number of Distributed Memory Interfaces configured in an Ncore interconnect			
Release Info	Status	Effective version	Visibility	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 2-4: NUMBER OF SNOOP FILTERS

Parameter Name	Number of SFs			
Value	Data Type	Architecture	Release	Default
			Min: 1 Max: 16	N/A
Constraint/Dependency				
Customer Description	Number of Snoop Filters			
Engineering Description	Total number of Snoop Filters configured in an Ncore interconnect			
Release Info	Status	Effective version	Visibility	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 2-5: NUMBER OF DISTRIBUTED VIRTUAL MEMORY SYSTEM ENGINES

Parameter Name	Number of DVEs			
Value	Data Type	Architecture	Release	Default
			Min: 1 Max: 1	N/A
Constraint/Dependency				
Customer Description	Number of DVEs			
Engineering Description	Total number of distributed virtual memory system engines configured in an Ncore interconnect			
Release Info	Status	Effective version	Visibility	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 2-6: NUMBER OF DISTRIBUTED COHERENCY ENGINES

Parameter Name	Number of DCEs			
Value	Data Type	Architecture	Release	Default
			Min: 1 Max: 16	N/A
Constraint/Dependency				
Customer Description	Number of DCEs			
Engineering Description	Total number of distributed coherency engines configured in an Ncore Interconnect			
Release Info	Status	Effective version	Visibility	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 2-7: NUMBER OF DISTRIBUTED IO INTERFACES

Parameter Name	Number of DIIs			
Value	Data Type	Architecture	Release	Default
			Min: 1 Max: 16	N/A
Constraint/Dependency				
Customer Description	Number of DIIs			
Engineering Description	Total number of distributed IO interfaces configured in an Ncore Interconnect			
Release Info	Status	Effective version	Visibility	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 2-8: NUMBER OF GIU

Parameter Name	nGIUs			
Value	Data Type	Architecture	Release	Default
			Min: 0 Max: 4	N/A
Constraint/Dependency	Can only be configured more than 0 when nDies parameter in chapter D2D is bigger than 1			
Customer Description	Number of GIUs			
Engineering Description	Total number of GIUs,			
Release Info	Status	Effective version	Visibility	
	Preview	3.8	Engineering	
Change History				

Field Code Changed

3. System User Settable Parameters

Commented [BM1]: Add wFunitId

Commented [BM2R1]: Add wNunitid

3.1. Project name and RTL prefix parameters

TABLE 3-1: PROJECTNAME PARAMETER

Parameter Name	projectName			
Value	Data Type	Architecture	Release	Default
Constraint/Dependency				
Customer Description	Project Name.			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 3-2: USERTLPREFIX PARAMETER

Parameter Name	useRtlPrefix			
Value	Data Type	Architecture	Release	Default
	Boolean			FALSE
Constraint/Dependency				
Customer Description				
Engineering Description	Once it is turned on and a string is specified, the string will be added globally to all Ncore units' module names. This will uniquify Ncore module names if more than one Ncore is instantiated on an SoC.			
Release Info	Status	Effective version	Visibility	
	Active	3.6.3	User-GUI	
Change History				

Field Code Changed

3.2. System level connectivity parameters

As described in Section 3.4.5 Message connectivity and network mapping of NCore System Architecture specification for Ncore 3.8. NCore provides the mapping templates of Concerto C messages to CDTI network. User will choose one of them considering the tradeoff between performance and area/power dissipation. For the detail of each mapping, refer NCore System Architecture document.

We are supporting three options:

- Use two command networks and one data network
- Use three command networks and one data network
- Use four command networks and one data network

TABLE 3-3: COHERENTTEMPLATE PARAMETER

Parameter Name	coherentTemplate			
Value	Data Type	Architecture	Release	Default
	Enum	TwoCtrlOneDataTemplate, ThreeCtrlOneDataTemplate, FourCtrlOneDataTemplate	FourCtrlOneDataTemplate	FourCtrlOneDataTemplate

Constraint/Dependency	Control and data network options: TwoCtrlOneDataTemplate: Adds support for two control and a single data network. ThreeCtrlOneDataTemplate: Adds support for three control and a single data network. FourCtrlOneDataTemplate: Adds support for four control and a single data network.		
Engineering Description			
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	User-GUI
Change History	Four control network option is available since Ncore 3.6		

Field Code Changed

TABLE 3-4: NAIUPORTS PARAMETER

Parameter Name	nAiuPorts			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	*Min: 1* *Max: 16*	*Min: 1* *Max: 8*	1
Constraint/Dependency	Powers of two valid values are 1, 2, 4, 8 and 16; Ports need to be same			
Customer Description	Specifies the number of AIU that are grouped together. These AIUs must be identical.			
Engineering Description	The parameter applies to any Initiator AIU type in Ncore i.e. CAIU, NCAIU or multi ported NCAIU These set of AIUs are treated as a single group of AIUs and must be identical. This parameter is on top of nNativeInterfacePorts as shown in **Error! Reference source not found.**, here it shows as a mutliported NCAIU with two AXI ports specified by nNativeInterfacePorts and then 2 NCAIUs specified by nAiuPorts.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	User-GUI	
Change History				

Field Code Changed

TABLE 3-5: APRIMARYAIUPORTBITS PARAMETER

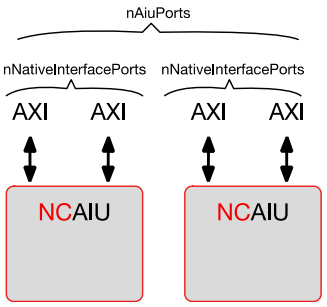
Parameter Name	aPrimaryAiuPortBits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	array of integers			
Constraint/Dependency	aPrimaryAiuPortBits depth depends on nAiuPorts parameter value it is limited to log2(nAiuPorts). Values must be address bits between Max address width minus 1 and cache line boundary address bit. For 64Bcache line it is 6. Values cannot overlap with the address bits used for cache sets/banks if an NCAIU contains cache for example proxy cache and interleaving bits used for nNativeInterfacePorts. Example aPrimaryAiuPortBits: [30, 9, 8, 6]			
Customer Description	Specify Address bits for port interleaving			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	User-GUI	
Change History				

Field Code Changed

TABLE 3-6: ASECONDARYAIUPORTBITS PARAMETER

Parameter Name	aSecondaryAiuPortBits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Array of strings			
Constraint/Dependency	aSecondaryAiuPortBits is an array of string, its depth depends on nAiuPorts parameter value it is limited to log2(nAiuPorts). The string represents a hexadecimal number one hot encoded. Bits selected here cannot be same as the bits in aPrimaryAiuPortBits. Example aSecondaryAiuPortBits: [“'h4000”, “'h0”, “'h0”, “'h800”]			
Customer Description				
Engineering Description	Not used in this release			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	User-GUI	
Change History				

Field Code Changed



69 FIGURE 3-1 NAIUPORTS PARAMETER

3.3. System address map parameters

In Ncore 3.x, we could have up to 24 configurable memory regions, and each memory region would be configured using registers. Please refer to memory address map session in system architecture specification for the details.

TABLE 3-7: NGPRA PARAMETER

Parameter Name	nGPRA			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	*Min: 2* *Max: 24*	*Min: 2* *Max: 24*	2[1]
Constraint/Dependency				
Customer Description	Number of general purpose address regions that the system can support			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

DCE is supporting fixed style interleaving, and the interleaving bits are configurable using the below parameters.

TABLE 3-8: DCEINTERLEAVINGPRIMARYBITS

Parameter Name	dceInterleavingPrimaryBits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Array of Integers			
Constraint/Dependency				
Customer Description	System directory primary select bits. Power of two number of DCEs : - N address bits other than bits 0 through 5 can be chosen. The cardinal values of these bits in the order of their ordinal positions are used to identify the DCE~~MI~~s to be accessed. Non-Power of two numbers of DCE : - Contains only one number which indicates the granularity of the interleaving i.e the size of the interleaved blocks of address. - Min value is 6 (64B) granularity - Max value is 12 (4kB)			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Adds non power of two interleaving to Ncore 3.8			

TABLE 3-9: DCEINTERLEAVINGSECONDARYBITS

[1] Minimum one coherent and one non-coherent space

Field Code Changed

Formatted: Left

Formatted: Font color: Auto

Formatted: Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

Field Code Changed

Parameter Name	dceInterleavingSecondaryBits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Array of Integers			
Constraint/Dependency				
Customer Description	The secondary bits are chosen on per primary bit bases. The bits within the set for a primary bit are combined and the primary bit with an Exclusive OR combination.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

3.4. System resiliency parameters

The resiliency feature in Ncore is optional, and when enabled, is implemented in addition to other configured Ncore features. The detail is described in Chapter Functional Safety in the NCore System Architecture Specification.

TABLE 3-10: RESILIENCE ON/OFF PARAMETER

Parameter Name	resilienceEnabled			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	FALSE	FALSE
Constraint/Dependency				
Customer Description	Enable resilience-related features in the Ncore system.			
Engineering Description	Auto-configured according to ASIL level configured by user			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	Engineering	
Change History	This option is visible in Maestro but user can't configure it, change the visibility to Engineering			

Field Code Changed

TABLE 3-11: SAFETYCONFIG PARAMETER

Parameter Name	safetyConfig			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	String	"NO_ASIL", "ASIL_A", "ASIL_B", "ASIL_D"	"NO_ASIL", "ASIL_A", "ASIL_B", "ASIL_D"	"NO_ASIL"
Constraint/Dependency				
Customer Description	This is a user visible parameter that turns on different levels of function safety protections. Once it is enabled (set true), the other two parameters such as resiliencyProtectionType and memoryProtectionType can be selectable.			
Engineering Description	This is a user visible parameter that turns on different levels of function safety protections.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	User-GUI	
Change History				

Field Code Changed

TABLE 3-12: DUPLICATION ENABLE PARAMETER

Parameter Name	duplicationEnabled			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	FALSE	FALSE
Constraint/Dependency				
Customer Description	Enable unit duplication for all Ncore units only. Memories and interconnect logic are not duplicated; they may be protected separately			
Engineering Description	Auto-configured according to ASIL level configured by user			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	Engineering	
Change History	This option is visible in Maestro but user can't configure it, change the visibility to Engineering			

Field Code Changed

This capability enables a designer to source or terminate data protection signals on selected external CAIU, IO-AIU, DMI, and DII interfaces.

TABLE 3-13: NATIVE INTERFACE PROTECTION PARAMETER

Parameter Name	nativeIntfProtEnabled			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	FALSE	FALSE
Constraint/Dependency				
Customer Description	Enable capability to add protection on native Ncore interfaces. This adds an empty Verilog module with specified signals at the interface. Protection logic can be added in this Verilog module.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	Engineering	
Change History				

Field Code Changed

The checker component receives one to four cycles delayed version of the same inputs as the functional component, which is decided by this parameter. The safety checker module receives the functional component outputs and delays them by one to four cycles, then compares them with the checker component outputs. Any discrepancy is considered a fault. Faults detected are logged and reported to the fault controller as mission fault. Once detected, the fault will remain logged inside the checker component until a BIST sequence clears it.

TABLE 3-14: INTERUNITDELAY PARAMETER

Parameter Name	interUnitDelay			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 4	Min: 1 Max: 4	*1*
Constraint/Dependency				
Customer Description	Delay between functional unit and delay unit. Delay can be specified in number of clock cycles.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	User-GUI	
Change History				

Field Code Changed

TABLE 3-15: RESILIENCYPROTECTIONTYPE PARAMETER

Parameter Name	resiliencyProtectionType			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		'"NONE"', '"PARITY"', '"SECDED"	'"NONE"', '"PARITY"', '"SECDED"	None
Constraint/Dependency				
Customer Description	Interconnect protection type. Both data and control header will be protected. Available options are: NONE: no protection. PARITY: Error detection, parity protection. SECDED: Single bit error correction and double bit error detection, ECC protection.			

Engineering Description	This parameter affects CDTI protection only.		
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.4	User-GUI
Change History			

Field Code Changed

111

TABLE 3-16: MEMORYPROTECTIONTYPE PARAMETER

Parameter Name	memoryProtectionType			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	String	'"NONE"', '"PARITY"', '"SECDED"	'"NONE"', '"PARITY"', '"SECDED"	None
Constraint/Dependency				
Customer Description	Memory protection type. Available options are: NONE: no protection. PARITY: Error detection, parity protection. SECDED: Single bit error correction and double bit error detection, ECC protection.			
Engineering Description	This parameter affects CDTI protection only.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	User-GUI	
Change History				

Field Code Changed

114 TABLE 3-17: FNDISABLERESILIENCYBISTDEBUGPIN PARAMETER

Parameter Name	fnDisableResiliencyBistDebugPin			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 1	Min: 0 Max: 1	0
Constraint/Dependency				
Customer Description	When set removes BIST and trace & debug disable pin.			
Engineering Description	When set removes BIST and trace & debug disable pin. This parameter is always on for the Automotive configuration			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	Engineering	
Change History				

Field Code Changed

3.5. System error parameters

This parameter configures timeout counter size in each module. We have additional register to configure the maximum size at run time, and the run time value should be less than or equal to this parameter value.

TABLE 3-18: TIMEOUTTHRESHOLDPARAMETER

Parameter Name	timeoutThreshold			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 2147483647	Min: 1 Max: 2147483647	16384
Constraint/Dependency				
Customer Description	Time out threshold value. This specifies number of clock cycles within which a transaction must complete in an NCORE system. The value specified is at 4096 clock cycle granularity.			
Engineering Description	From MAES-7574			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

Field Code Changed

TABLE 3-19: MEMORYPROTECTIONTYPE PARAMETER

Parameter Name	memoryProtectionType			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	String	'"NONE"', '"PARITY"', '"SECDED"	None	None
Constraint/Dependency				
Customer Description	Protection type for all memories in the Ncore system. Available options are: NONE : no protection. PARITY : Error detection, parity protection. SECDED : Single bit error correction and double bit error detection, ECC protection. SRAM memory type does not support memoryProtectionType ==NONE. If the memory is configured as FLOP, then NONE is supported.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	User-GUI	
Change History				

Field Code Changed

3.6. System QoS parameters

129 This parameter would enable starvation and aging arbitration in the skid buffer or OTT entry in AIU, DCE, and DMI.

TABLE 3-20: QOSENABLED PARAMETER

Parameter Name	qosEnabled			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	False
Constraint/Dependency				
Customer Description	Enable QoS support			
Engineering Description	When in a multi-die system it applies to every chiplet in the assembly.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

132

TABLE 3-21: QOSMAP PARAMETER

Parameter Name	qosMap			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	List of String	"qosMap": ["16'hc000", "16'h3000", "16'h0c00", "16'h0300", "16'h00c0", "16'h0030", "16'h000c", "16'h0003"],	"qosMap": ["16'hc000", "16'h3000", "16'h0c00", "16'h0300", "16'h00c0", "16'h0030", "16'h000c", "16'h0003"],	"qosMap": ["16'h0003"],
Constraint/Dependency	This parameter is available only when the parameter qosEnable is true			
Customer Description	4 bit Native interface QoS value map to 3 bit priority used inside an Ncore. **Value 0 is the highest priority and value 7 is the lowest priority**.			
Engineering Description	The 4-bit QoS value for an incoming native transaction is mapped to one of 8 QoS buckets (3-bit value priority field) using this parameter. The mapped values are being used for the QoS arbitration in skid buffer and OTT entries in AIU, DCE, and DMI. Here is the mapping: "qosMap": ["16'hc000", native QoS 15 – 14 maps to → priority 0 "16'h3000", native QoS 13 – 12 maps to → priority 1 "16'h0c00", native QoS 11 – 10 maps to → priority 2 "16'h0300", native QoS 9 – 8 maps to → priority 3 "16'h00c0", native QoS 7 – 6 maps to → priority 4 "16'h0030", native QoS 5 – 4 maps to → priority 5 "16'h000c", native QoS 3 – 2 maps to → priority 6 "16'h0003" native QoS 1 – 0 maps to → priority 7],			
Release Info	**Status**	**Effective version**	**Visibility**	

	Active	3.2	User-GUI
Change History			

Field Code Changed

TABLE 3-22: QOSEVENTTHRESHOLD PARAMETER

Parameter Name	qosEventThreshold			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 8192	Min: 1 Max: 8192	16
Constraint/Dependency	This parameter is available only when the parameter qosEnable is true			
Customer Description	QoS starvation threshold. Maximum number of high priority requests that can bypass a lower priority request.			
Engineering Description	Ncore 3 implements a single global counter as a time reference for starvation detection. Once the counter reaches a programmable threshold, an overflow bit in all active entries is set and the counter restarts. All transactions which had the overflow bit set at the time of the counter expiration will be considered starved and will be scheduled ahead of all non-starved transactions.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

3.7. System level debug parameters

Trace accumulate block (which is accumulate traces from AIU, DMI, and DII) is present only in DVE and the main functionality is to accumulate incoming trace DTWs from different NCore capture units. The capture buffer is sized based on the parameter nMainTraceBufSize

For the trace entries, user could configure as SRAM using user interface.

TABLE 3-23: NMAINTRACEBUFSIZE PARAMETER

Parameter Name	nMainTraceBufSize			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 32 Max: 4096	Min: 32 Max: 1024	64
Constraint/Dependency				
Customer Description	Number of trace entries in the buffer			
Engineering Description	Number Debug DTW entries the trace buffer can hold. The actual depth of the trace buffer may be larger depending on the data width for the design. Each debug DTW can be max 64 bytes.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 3-24: NTRACEREGISTERS PARAMETER

Parameter Name	nTraceRegisters			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 8	Min: 1 Max: 4	1
Constraint/Dependency	Minimum 1 is required			
Customer Description	Number of trace trigger configuration register sets. Each set of register can enable a trace condition.			
Engineering Description	All AIUs, DMIs and DIIs shall support trace capturing capability. The block snoops SMI interface and captures messages that have the TraceMe field set. The incoming transaction on the interfaces is compared with the trace CSR settings, if there is a match the transaction is marked to be traced. Multiple number of CSR sets can be present as specified at build time by the parameter nTraceRegisters. The capture block has a capture buffer that is sized based on the parameter nUnitTraceBufSize, this parameter specifies the number of 64-byte entries in the buffer. For the trace entries, user could configure as SRAM using user interface.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 3-25: NUNITTRACEBUFSIZES PARAMETER

Parameter Name	nMainTraceBufSize			
Value	**Data Type**	**Architecture**	**Release**	**Default**

	Integer	Min: 8 Max: 32	Min: 8 Max: 16	8
Constraint/Dependency	Allowable size are power of 2			
Customer Description	Number of trace entries in each Ncore Unit. Each entry is 64 bytes			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 3-26: FNDEBUGAPBENABLE PARAMETER

Parameter Name	fnDebugAPBEnable			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 1	Min: 0 Max: 1	1
Constraint/Dependency				
Customer Description	When set enables an APB slave port on the CSR network. This port is expected to be used for on chip debug purposes only.			
Engineering Description	To enable debug of a hung Ncore system a slave APB port must be added to the CSR network that can access all the Ncore CSRs. At top level this port signals must be "<prefix>_debug_apb_<rest of the signal name>". Following APB port restrictions apply • Fixed data bus width 32 bits • Fixed address bus width of 20 bits • Fixed access size of 4 bytes • All access are 4 byte aligned. This port is expected to be used for debug only, if same register is accessed concurrently via this debug APB port and the internal Ncore CSR accesses then the effect on the CSR is undefined. Ncore does not guarantee any ordering between the two access.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	User-GUI	
Change History				

Field Code Changed

3.8. System level physical parameters

159

TABLE 3-27: SYNCDEPTH PARAMETER

Parameter Name	syncDepth			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid values	2, 3, 4	2,3,4	2
Constraint/Dependency				
Customer Description	The depth of the synchronizers used for signals that cross domains for metastability reasons. This is only for sym_async_adapter. FIFO depth of the chi_async_adapter would be calculate considering credit at the CHI interface link.			
Engineering Description	Circular FIFO depth of the sym_async_adpater would be derived by this system configuration value • syncDepth: 2 → circular fifo depth of sym_async_adapter: 8 • syncDepth: 3 → circular fifo depth of sym_async_adapter: 10 • syncDepth: 4 → circular fifo depth of sym_async_adapter: 12			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

162

3.9. System engineering parameters

165 Engineering Only parameter should not be visible to the customer.

TABLE 3-28: ASSERTIONENABLE PARAMETER

Parameter Name	assertionEnable			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean			FALSE
Constraint/Dependency				
Customer Description				
Engineering Description	Enable HW assertions			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

168

TABLE 3-29: ENGVERID PARAMETER

Parameter Name	EngVerId			
Value	**Data Type**	**Architecture**	**Release**	**Default**
Constraint/Dependency				
Customer Description				
Engineering Description	Refer to Engineering version id *32 bits in total,* 19 bits are reserved for MPF hash (every time gen_collateral command is issued, copy mpf is saved. After that 128 bits MD5 hash is used to get hash and last 19 bits used for engVerId) 13 bits are reserved for CHIP_ID from a license file. Example if CHIP_ID is 1001, engVerId looks like xxxxxxxxxxxxxxxxxx0001111101001, where x is a mpf hash.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

Field Code Changed

171 TABLE 3-30: IMPLVERID PARAMETER

Parameter Name	ImplVerId			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer			
Constraint/Dependency				
Customer Description				
Engineering Description	Refer to Engineering version id 16 bits to store Ncore version, Format : {4'd, 4'd, 4'd, 4'd}, 4bits per digit. For example Ncore 3.6.2.6 => {16'h3626} or { 4'd3, 4'd6, 4'd2, 4'd6 }			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	

Field Code Changed

Field Code Changed

Change History	

3.10. RISC-V related parameters

TABLE 3-31: NODVM PARAMETER

Parameter Name	noDVM			
Value	Data Type	Architecture	Release	Default
	Boolean			FALSE
Constraint/Dependency				
Customer Description	Disable DVM related functionality throughout an Ncore and does not reserve any DVM related credits such as DVM snoop credits across all Ncore units			
Engineering Description	Disable DVM related functionality throughout an Ncore and does not reserve any DVM related credits such as DVM snoop credits across all Ncore units			
Release Info	Status	Effective version	Visibility	
	Active	3.6.4	User-GUI	
Change History				

Field Code Changed

3.11. D2D system parameters

TABLE 3-32: NDIES PARAMETERS

Parameter Name	nDies			
Value	Data Type	Architecture	Release	Default
	Valid Values	{1,2,3,4}	{1,2,3,4}	2
Constraint/Dependency				
Customer Description	Number of die in the system. Set to 1 for single die configuration. Max value is 4.			
Engineering Description	This is the number of die in the system. For single dies configuration, the value is 1 for multi dies the value is greater than 1 and maximum 4.			
Release Info	Status	Effective version	Visibility	
	Preview	3.8	User-GUI	
Change History				

Field Code Changed

TABLE 3-33: NREMOTESNPCREDITS PARAMETERS

Parameter Name	nRemoteSnpCredits			
Value	Data Type	Architecture	Release	Default
	Integer	Min : 8, Max : 127	Min : 8, Max : 127	32
Constraint/Dependency	Must be greater than max(nDCEs).			

Customer Description	This is the number of snoops that a chiplet can send to any other remote chiplet. The number of buffers created in each AIU will be n**RemoteSnpCredits, This means that each DCE will have** $[nRemoteSnpCredit/nDces]$		
Engineering Description	This is the number of snoops that a chiplet can send to any other remote chiplet. The number of buffers created in each AIU will be n**RemoteSnpCredits, This means that each DCE will have** $[nRemoteSnpCredit/nDces]$		
Release Info	**Status**	**Effective version**	**Visibility**
	Preview	3.8	User-GUI
Change History			

Field Code Changed

Table 3-34: GlobalLargestNExclusiveThreads parameters

Parameter Name	GlobalLargestNExclusiveThreads			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 32	Min: 0 Max: 32	0
Constraint/Dependency	Must be larger or equal to every AIU nProc in the system. Only exists in multi die systems.			
Customer Description	Specifies the largest number of processors connected to an AIU in the entire system. It is used to size exclusive monitor entries			
Engineering Description	Specifies the largest number of processors connected to an AIU in the entire system. This is used to size exclusive monitor entries This parameter is only for multi-die system, not available for single die system.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Preview	3.8	User-GUI	
Change History				

Field Code Changed

TABLE 3-35: NGLOBALCACHINGAGENTS PARAMETERS

Parameter Name	nGlobalCachingAgents			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 128	Min: 0 Max: 128	32
Constraint/Dependency	This value needs to be equal to the number of fully coherent agents in the entire system.			
Customer Description	This parameter is used to size the number of agent tracked by the directory.			
Engineering Description	This parameter is used to size the number of agent tracked by the directory. It is derived for single die systems and set by the user at assembly definition step for multi-die systems. This parameter is only for multi-die system, not available for single die system.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Preview	3.8	User-GUI	
Change History				

Field Code Changed

4. System non-user visible / derived parameters

4.1. System parameters

TABLE 4-1 NDCES PARAMETER

Parameter Name	nDces			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	~~*Integer*~~*Enum*	1,2,3,4,6,8,12,16~~1-16~~	1,2,3,4,6,8,12,16~~1-16~~	NA
Constraint/Dependency	Number of DCEs in the system.			
Customer Description	NA			
Engineering Description	Number of DCEs in the system			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3	Engineering	
Change History	Adds 3.6 and 16 in Ncore 3.8.			

Field Code Changed

TABLE 4-2 NDIIS PARAMETER

Parameter Name	nDiis			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	1-16	1-16	NA
Constraint/Dependency	Number of DIIs in the system.			
Customer Description	NA			
Engineering Description	Number of DIIs in the system			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3	Engineering	
Change History				

Field Code Changed

TABLE 4-3 NDMIS PARAMETER

Parameter Name	nDmis			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	1-16	1-16	NA
Constraint/Dependency	Number of DMIs in the system.			
Customer Description				
Engineering Description	Number of DMIs in the system.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3	Engineering	
Change History				

Field Code Changed

TABLE 4-4 NAIUS PARAMETER

Parameter Name	nAius			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	1-32	1-32	NA
Constraint/Dependency	Number of AIUs in the system			
Customer Description				
Engineering Description	Number of AIUs in the system			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3	Engineering	
Change History				

Field Code Changed

TABLE 4-5 NNCAIUS PARAMETER

Parameter Name	nNCAius			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	0-16	0-16	NA
Constraint/Dependency	Number of NCAius in the system			
Customer Description	NA			
Engineering Description	Number of NCAius in the system			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3	Engineering	
Change History				

Field Code Changed

TABLE 4-6 NCAIUS PARAMETER

Parameter Name	nCAius			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	1-16	1-16	NA
Constraint/Dependency	Number of Caius in the system			
Customer Description	NA			
Engineering Description	Number of Caius in the system			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3	Engineering	
Change History				

Field Code Changed

TABLE 4-7 ~~NCAIUS~~ NDVES PARAMETER

Parameter Name	nDves			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	1	1	NA
Constraint/Dependency	Number of DVE in the system			
Customer Description	NA			
Engineering Description	Number of DVE in the system			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3	Engineering	

Field Code Changed

Change History	

TABLE 4-8 NPROXYCACHES PARAMETER

Parameter Name	nProxyCaches			
Value	Data Type	Architecture	Release	Default
	Integer			
Constraint/Dependency	Number of proxy caches in the system			
Customer Description				
Engineering Description	Number of proxy caches in the system			
Release Info	Status	Effective version	Visibility	
	Active	3	Engineering	
Change History				

Field Code Changed

TABLE 4-9 NCHIAAIUS PARAMETER

Parameter Name	nChiaAius			
Value	Data Type	Architecture	Release	Default
	Integer	0	0	NA
Constraint/Dependency	Number of CHI A in the system			
Customer Description				
Engineering Description	Number of CHI A in the system			
Release Info	Status	Effective version	Visibility	
	Deprecated	3	Engineering	
Change History				

Field Code Changed

TABLE 4-10 NCHIBAIUS PARAMETER

Parameter Name	nChibAius			
Value	Data Type	Architecture	Release	Default
	Integer	0-16	0-16	NA
Constraint/Dependency	Number of CHI B in the system			
Customer Description				
Engineering Description	Number of CHI B in the system. This is a sub-type of Caius.			
Release Info	Status	Effective version	Visibility	
	Active	3	Engineering	
Change History				

Field Code Changed

TABLE 4-11 NACEAIUS PARAMETER

Parameter Name	nAceAius			
Value	Data Type	Architecture	Release	Default
	Integer	0-16	0-16	NA

Constraint/Dependency	Number of ACE in the system		
Customer Description			
Engineering Description	Number of ACE in the system. This is a sub-type of Caius.		
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3	Engineering
Change History			

Field Code Changed

TABLE 4-12 NACELITEAIUS PARAMETER

Parameter Name	nAceLiteAius			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	0-16	0-16	NA
Constraint/Dependency	Number of ACE-Lite in the system			
Customer Description				
Engineering Description	Number of ACE-lite in the system. This is a sub-type of NCaius.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3	Engineering	
Change History				

Field Code Changed

TABLE 4-13 NACELITEEAIUS PARAMETER

Parameter Name	nAceLiteEAius			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	0-16	0-16	NA
Constraint/Dependency	Number of ACE5-Lite in the system			
Customer Description				
Engineering Description	Number of ACE5-lite in the system. This is a sub-type of NCAius.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3	Engineering	
Change History				

Field Code Changed

TABLE 4-14 NAXIAIUS PARAMETER

Parameter Name	nAceLiteAius			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	0-16	0-16	NA
Constraint/Dependency	Number of AXI in the system (AXI4 and AXI5)			
Customer Description				
Engineering Description	Number of AXI in the system (AXI4 and AXI5). It is a sub-type of NCAius.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3	Engineering	
Change History				

Field Code Changed

TABLE 4-15 WFUNITID PARAMETER

Parameter Name	wFUnitId			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer			
Constraint/Dependency	Single Die : Log2ceil(nAius + nDmis + nDiis + nDces + nDves + nGius) Multie Die : 8.			
Customer Description				
Engineering Description	Width of the FUnitId field which is a unique identifier associated to Ncore unit			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3	Engineering	
Change History				

Field Code Changed

TABLE 4-16 WNUNITID PARAMETER

Parameter Name	wNUnitId			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer			
Constraint/Dependency	Log2ceil(max(nAius, nDmis, nDiis, nDces, nDves, nGius))			
Customer Description				
Engineering Description	Width of the NUnitId which identifies a unit within its own class of unit. It indexes units of the same type.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3	Engineering	
Change History				

Field Code Changed

4.2. GIU interleaving parameters

TABLE 4-17: REMOTEINTERLEAVINGOBJECT OBJECT REPRESENTING THE INTERLEAVING OF GIUS

Parameter Name	system.GIUInterleaveInfo.RemoteInterleavingObject			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Multi-dimensional array of int			
Constraint/Dependency	RemoteLinkInterleavingObject [MyAssemblyId][MyChipletId][RemoteChipletId][IG][InterleavingFunction(Addr)] = LinkId			
Customer Description				
Engineering Description	This multi-dimensional arrays allows hardware to obtain a LinkId from an address when GIU interleaving is used. It is derived from customer input to set the topology and the GIU interleaving. Maestro			
Release Info	**Status**	**Effective version**	**Visibility**	
	Preview	3.8	Engineering	
Change History				

Field Code Changed

TABLE 4-18: GIU2WIFV GIU 2 WAY INTERLEAVING FUNCTION

Parameter Name	system.GIUInterleaveInfo.giu2WIFV			
Value	Data Type	Architecture	Release	Default
	object			
Constraint/Dependency	{"PrimaryBits" : [<int>], "SecondaryBits : [<string>] }, length of the PrimaryBits and SecondaryBits array is 1			
Customer Description				
Engineering Description	This object contains the 2 way interleaving function which can be used for interleaving GIUs			
Release Info	Status	Effective version	Visibility	
	Preview	3.8	Engineering	
Change History				

TABLE 4-19: GIU3WIFV GIU 3 WAY INTERLEAVING FUNCTION

Parameter Name	system.GIUInterleaveInfo.giu3WIFV			
Value	Data Type	Architecture	Release	Default
	object			
Constraint/Dependency	Contains the 3 way interleaving function defined by the customer for GIU interleaving. Structure of the object is : {"PrimaryBits" : [<int>], "SecondaryBits : [<string>] }, length of the PrimaryBits and SecondaryBits array is 2			
Customer Description				
Engineering Description	This object contains the 3 way interleaving function which can be used for interleaving GIUs			
Release Info	Status	Effective version	Visibility	
	Preview	3.8	Engineering	
Change History				

TABLE 4-20: GIU3WIFV GIU 4 WAY INTERLEAVING FUNCTION

Parameter Name	system.GIUInterleaveInfo.giu4WIFV			
Value	Data Type	Architecture	Release	Default
	object			
Constraint/Dependency	Contains the 4 way interleaving function defined by the customer for GIU interleaving. Structure of the object is : {"PrimaryBits" : [<int>], "SecondaryBits : [<string>] }, length of the PrimaryBits and SecondaryBits array is 2			
Customer Description				
Engineering Description	This object contains the 4 way interleaving function which can be used for interleaving GIUs			
Release Info	Status	Effective version	Visibility	
	Preview	3.8	Engineering	
Change History				

TABLE 4-21: GIU8WIFV GIU 8 WAY INTERLEAVING FUNCTION

Parameter Name	system.GIUInterleaveInfo.giu8WIFV			
Value	Data Type	Architecture	Release	Default
	object			
Constraint/Dependency	Contains the 8 way interleaving function defined by the customer for GIU interleaving. Structure of the object is : {"PrimaryBits" : [<int>], "SecondaryBits : [<string>] }, length of the PrimaryBits and SecondaryBits array is 3			
Customer Description				
Engineering Description	This object contains the 8 way interleaving function which can be used for interleaving GIUs			
Release Info	Status	Effective version	Visibility	
	Preview	3.8	Engineering	
Change History				

TABLE 4-22: GIU16WIFV GIU 16 WAY INTERLEAVING FUNCTION

Parameter Name	system.GIUInterleaveInfo.giu16WIFV			
Value	Data Type	Architecture	Release	Default
	object			
Constraint/Dependency	Contains the 16 way interleaving function defined by the customer for GIU interleaving. Structure of the object is : {"PrimaryBits" : [<int>], "SecondaryBits : [<string>] }, length of the PrimaryBits and SecondaryBits array is 4			
Customer Description				
Engineering Description	This object contains the 16 way interleaving function which can be used for interleaving GIUs			
Release Info	Status	Effective version	Visibility	
	Preview	3.8	Engineering	
Change History				

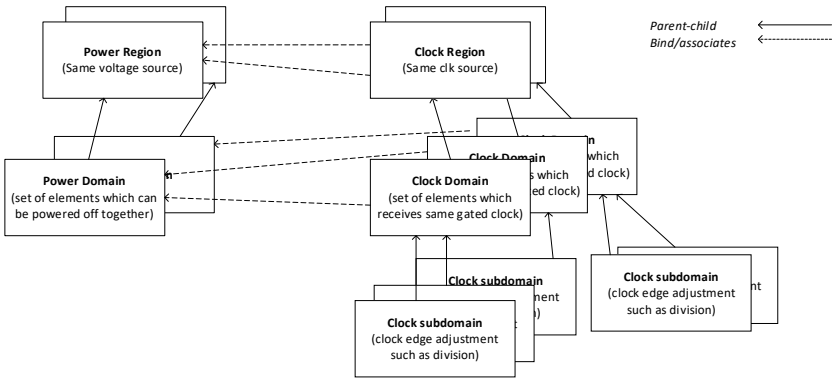
5. *Power and Clock User Settable Parameters*

Clocking and Power are defined in terms of regions, domains, and sub domains:

- A power region represents a group of elements that run off a power supply that is driven by one power source.
- A clock region represents a group of elements that are clocked by single clock.
- A power domain represents a group of elements whose power can be turned on and off.
- A clock domain represents a group of elements whose clock can be turned on and off.
- There are no power sub domains.

A clock sub domain is a group of elements in a clock domain whose clocks can be turned on and off dynamically as a part of logic function (clock divider). There are no clock dividers supported in Ncore 3.2.

FIGURE 5-1: POWER AND CLOCK DOMAIN DEFINITION FOR NCORE 3.6



Three levels of clock gating are supported since Ncore 3.x:

- The first level clock gating is enabled by synthesis tools, for example Synopsys Design Compiler.
- A second level of clock gating will be inserted, one per Ncore unit. This level gates the clock for the complete unit when no active transactions are within that unit. This is enabled by "unitClockGating" parameter of clock region.
- A third level of clock gating can be achieved by using the q-channel.
  This is enabled by "Gating" parameter of clock domain.

By default, clock subdomains are async with other clock domains, but NCore 3.x would need to allow clock domains which share the same clock root to be explicitly defined to be synchronous with each other.
The reason is to allow the Ncore units to be gated without gating the CSR network and potentially other networks so that those messages do not get accidentally trapped. In this case user would define two clock domains which were synchronous to each other, with one of them being dynamic and the other not. The

dynamic clock would be used for connecting the Ncore units while the non-dynamic clock would be associated with other components such as the CSR network.

To support the above, the followings are applied:

- Sub domain update:
  - There will be one single clock subdomain per clock domain. There are no clock dividers.
  - Only allow one clock subdomain per domain
- Clock domain update:
  - Will allow clock domains which share the same clock root and explicitly defined to be synchronous.
  - Async adapter would not be inserted at synchronous clock boundary.

Restrictions:

- Supports only single power domain.
  - NCore 3.x does NOT provide a UPF file to the customer that describes where the level shifters and clamping cells (and the associated clamping values) for signals that cross between power domains.
  - It is user's responsiblity to support multiple power domain using clock region/domain capability.
- Does NOT support retention mode.
- Does NOT support auto-wakeup, that is, QACTIVE during the powered down state will not assert indicating a request to the PMU (User's power control unit) to wake up

Detach process (SysCoReq/SysCoAck):

Before NCore unit clock gating, attach and Detach to the coherent domain should be performed by SysCo/SysAck. This will be controlled by CPU events or CSR interface setting.

TABLE 5-1: PARAMETER RELATED WITH **CLOCK REGION**: FREQUENCY

Parameter Name	Frequency (MHz)			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 2000	Min: 1 Max: 2000	500
Constraint/Dependency				
Customer Description				
Engineering Description	NCore 3.7 supports 2GHz for IOAIUp as a maximum frequency. This should be visible only for that ranges.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 5-2: PARAMETER RELATED WITH **CLOCK REGION**: UNITCLOCKGATING

Parameter Name	unitClockGating			
Value	**Data Type**	**Architecture**	**Release**	**Default**

	Valid Values	True, False	FALSE	FALSE
Constraint/Dependency				
Customer Description	When the parameter is true, then the blocks in the corresponding clock region will insert clock gating based on its internal and the state of the interfaces connected to it.			
Engineering Description	Not all blocks will insert clock gates when this parameter is set to true. For instance, blocks sym_async_adapter and sym_rate_adapter do not insert clock gating in response to this parameter.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 5-3: PARAMETER RELATED WITH **CLOCK DOMAIN**: GATING

Parameter Name	Gating			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	always_on, external	always_on	always_on
Constraint/Dependency				
Customer Description	Specify 'always_on' if no gating applied, or 'external' if gated control logic is applied externally			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

315

6. Memory Map User Settable Parameters

Commented [BM3]: Review, this seems outdated.

Ncore 3.x address map is categorized into three main spaces:

- Ncore Register Space (NRS): This address space is reserved by Ncore 3 architecture for mapping Control and Status registers belonging to Ncore 3 units. Each Ncore 3 unit's registers map within a single 4 KB block of address space.
- General Purpose Address Space (GPAS): The remaining address space is available for general purpose use. It may contain multiple system memory or peripheral storage ranges. General purpose address space may be comprised of one or regions of type system memory or peripheral storage. The system memory regions can be accessed coherently or non-coherently.
- Boot Region (BR): Ncore 3 permits the SoC system to identify a contiguous aligned block of address space for the boot code to reside in. The boot code might be accessed by a processor during the system boot process when no other addressmapping might be valid. The type of storage occupied by the Boot Space can be system memory or peripheral memory.

Listed below are parameters used to configurable memory space :

TABLE 6-1: PARAMETER RELATED WITH **CSR REGION**: MEMORYBASE

Parameter Name	memoryBase			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values		0x0	0x2e800000
Constraint/Dependency				
Customer Description	Specify CSR region base address. This address must be aligned to the size specified. For Multi die configuration this parameter is common for all dies. The address region assigned to each dies is memoryBase + ChipletId x MemorySize			
Engineering Description	This is an assembly level parameter.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 6-2: PARAMETER RELATED WITH **CSR REGION**: MEMORYSIZE

Parameter Name	memorySize			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	1MB	1MB	1MB
Constraint/Dependency				
Customer Description	CSR sized is fixed as 1MB from NCore 3.2			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 6-3: PARAMETER RELATED WITH **BOOT REGION:** MEMORYBASE

Parameter Name	memoryBase			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values		0x0	0x0
Constraint/Dependency				
Customer Description	Specify boot region base address. This address must be aligned to the size specified.			
Engineering Description	Must be aligned to 4KB. This parameter can be different across dies in multi-die configuration. If a chiplet is instanciated multiple times in a configuration, it will have the same MemoryBase for all the instances. In other word this is a chiplet level parameter.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 6-4: PARAMETER RELATED WITH **BOOT REGION:** MEMORYSIZE

Parameter Name	memorySize			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	Min: 4KB, Max: 4096KB	Min: 4KB, Max: 4096KB	16K
Constraint/Dependency				
Customer Description	Specifies the size of boot region. Minimum is 4KB and must be a power of two.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 6-5: PARAMETER RELATED WITH **BOOT REGION:** MG_REF

Parameter Name	mg_ref			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values			
Constraint/Dependency				
Customer Description	Specify the DMI interleave group associated with the boot region.			
Engineering Description	If mg_ref is specified, channel_ref cannot be specified			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 6-6: PARAMETER RELATED WITH **BOOT REGION:** CHANNEL_REF

Parameter Name	channel_ref			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values			
Constraint/Dependency				
Customer Description	Specify the DII group associated with the boot region.			
Engineering Description	If channel_ref is specified, mg_ref cannot be specified			
Release Info	**Status**	**Effective version**	**Visibility**	

	Active	3.2	User-GUI
Change History			

Field Code Changed

Dynamic Memory Group:

Dynamic memory group is to define GPARS (Number of GPARS is configured by **Error! Reference source not found.**). For each dynamic memory group, the target DMIs are bounded. The base and size are configured for each group. If we bound more than two DMIs into one dynamic memory group, we need interleaving granularity, which is configured by Interleaving Functions.

- Starting with Ncore 3.2, user could bound only 1, 2, 4, 8, and 16 DMIs into one dynamic memory group (=MIG).
- If a dynamic memory group have more than 2 DMIs, we need to define the interleaving function (=MIF) for each DMI. The below tables are user settable parameters to define interleave function.
- The maximum number of interleaving functions (=interleaving granularity) is 2 Starting with Ncore 3.2.

Please see Chapter Address Map Specification in System Architecture spec for the detail.

The primaryInterleavingBits are used to specify interleaving function per each interleaving group.

TABLE 6-7: PRIMARYINERLEAVINGBITONE

Parameter Name	primaryInterleavingBitOne			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 63	Min: 0 Max: 63	0
Constraint/Dependency				
Customer Description				
Engineering Description	This primaryInterleavingBitOne is per MIF.			
Release Info	**Status**	**Effective version**	**Visibility**	
	~~Active~~Deprecated	3.2	User-GUI	
Change History	Removed in 3.8			

Field Code Changed

TABLE 6-8: PRIMARYINERLEAVINGBITTWO

Parameter Name	primaryInterleavingBitTwo			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 63	Min: 0 Max: 63	0
Constraint/Dependency				
Customer Description				
Engineering Description	This primaryInterleavingBitTwo is per MIF.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Deprecated~~Active~~	3.2	User-GUI	
Change History	Removed in 3.8			

Field Code Changed

TABLE 6-9: PRIMARYINERLEAVINGBITTHREE

Parameter Name	primaryInterleavingBitThree			
Value	Data Type	Architecture	Release	Default
	Integer	Min: 0 Max: 63	Min: 0 Max: 63	0
Constraint/Dependency				
Customer Description				
Engineering Description	This primaryInterleavingBitThree is per MIF.			
Release Info	Status	Effective version	Visibility	
	DeprecatedActive	3.2	User-GUI	
Change History	Removed in 3.8			

Field Code Changed

366

TABLE 6-10: PRIMARYINERLEAVINGBITFOUR

Parameter Name	primaryInterleavingBitFour			
Value	Data Type	Architecture	Release	Default
	Integer	Min: 0 Max: 63	Min: 0 Max: 63	0
Constraint/Dependency				
Customer Description				
Engineering Description	This primaryInterleavingBitFour is per MIF.			
Release Info	Status	Effective version	Visibility	
	ActiveDeprecated	3.2	User-GUI	
Change History	Removed in 3.8			

Field Code Changed

369

TABLE 6-11: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.TWOWAYINTRLVFUNCV

Parameter Name	TwoWayIntrlvFuncV			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Object	{PrimaryBits : [], SecondaryBits: ["h0"]}	{PrimaryBits : [], SecondaryBits: "'h0"]}	{PrimaryBits : [6], SecondaryBits : "'h0"]}
Constraint/Dependency	Max 2 2 way interleaving functions. SecondaryBits are ***not*** user visible and should always be 'h0. PrimaryBits array length is 1 and takes integer between 6 and wAddr-1			
Customer Description	Selects the bit used for the two ways memory interleaving function. It takes one bit between 6 and address width -1. There is a maximum of two such functions and the bit used should not overlap with the bit used to generate the sets in the caches and the directory.			
Engineering Description	Two way interleaving function to be used inside Ncore Address map. The user visible name in the GUI should be "Primary InterleavingBit". If we ever support hashing we would need to allow the customer to also set the secondary bits.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.8	User-GUI	
Change History	Already present in earlier version but not documented this way			

Formatted: Font: Bold

TABLE 6-12: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.THREEWAYINTRLVFUNCV

Parameter Name	ThreeWayIntrlvFuncV			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Object	{PrimaryBits : [], SecondaryBits: ["h0"]}	{PrimaryBits : [], SecondaryBits: ["'h0"]}	{PrimaryBits : [6], SecondaryBits : ["'h0"]}
Constraint/Dependency	Max 2 3-way interleaving functions. Primary bit defines the granularity of the interleaving. SecondaryBits are not user visible. PrimaryBits array length is 1 and takes an integer between 6 and 12.			
Customer Description	Three-way interleaving function. Selects the granularity of the interleaving. - Min value is 6 (64B granularity) - Max value is 12 (4kB) There is a maximum of two such functions.			
Engineering Description	Three-way interleaving function to be used inside Ncore Address map. User visible name in the GUI should be "interleaving Granularity". If we ever support hashing this would need to instead allow the customer to set an array of primary and secondary bits			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.8	User-GUI	
Change History	New for 3.8			

TABLE 6-13: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.FOURWAYINTRLVFUNCV

Parameter Name	FourWayIntrlvFuncV			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Object	{PrimaryBits : [], SecondaryBits: ["h0","h0"]}	{PrimaryBits : [], SecondaryBits: ["'h0","h0"]}	{PrimaryBits : [6,7], SecondaryBits : ["'h0","h0"]}
Constraint/Dependency	Max 2 4-way interleaving functions. SecondaryBits are not user visible. PrimaryBits array length is 2 and takes integer between 6 and wAddr-1			
Customer Description	Selects the bit used for the four ways memory interleaving function. It takes two bits between 6 and address width -1. There is a maximum of two such functions and the bit used should not overlap with the bit used to generate the sets in the caches and the directory.			
Engineering Description	Four-way interleaving function to be sued inside Ncore Address map. "Primary Interleaving Bits (2bits)". If we ever support hashing we would need to allow the customer to also set the secondary bits.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.8	User-GUI	
Change History	Already present in earlier version but not documented this way			

TABLE 6-14: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.SIXWAYINTRLVFUNCV

Parameter Name	SixWayIntrlvFuncV			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Object	{PrimaryBits : [], SecondaryBits: ["h0"]}	{PrimaryBits : [], SecondaryBits: ["'h0"]}	{PrimaryBits : [6], SecondaryBits : ["'h0"]}
Constraint/Dependency	Max 2 6-way interleaving functions. Primary bit defines the granularity of the interleaving. SecondaryBits are not user visible. PrimaryBits array length is 1 and takes an integer between 6 and 12.			
Customer Description	Six ways interleaving function. Selects the granularity of the interleaving. - Min value is 6 (64B granularity) - Max value is 12 (4kB) There is a maximum of two such functions.			
Engineering Description	Three-way interleaving function to be used inside Ncore Address map. User visible name in the GUI should be "interleaving Granularity". If we ever support hashing this would need to instead allow the customer to set an array of primary and secondary bits			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.8	User-GUI	
Change History	New for 3.8			

TABLE 6-15: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.EIGHTWAYINTRLVFUNCV

Parameter Name	EightWayIntrlvFuncV			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Object	{PrimaryBits : [], SecondaryBits: ["h0","h0","h0"]}	{PrimaryBits : [], SecondaryBits: ["'h0","h0","'h0"]}	{PrimaryBits : [6,7,8], SecondaryBits : ["'h0","'h0","h0"]}
Constraint/Dependency	Max 2 8-way interleaving functions. SecondaryBits are not user visible. PrimaryBits array length is 3 and takes integer between 6 and wAddr-1			
Customer Description	Selects the bit used for the eight-way memory interleaving function. It takes three bits between 6 and address width -1. There is a maximum of two such functions and the bit used should not overlap with the bit used to generate the sets in the caches and the directory.			
Engineering Description	Four-way interleaving function to be sued inside Ncore Address map. "Primary Interleaving Bits (3 bits)". If we ever support hashing we would need to allow the customer to also set the secondary bits.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.8	User-GUI	
Change History	Already present in earlier version but not documented this way			

TABLE 6-16: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.TWELVEWAYINTRLVFUNCV

Parameter Name	TwelveWayIntrlvFuncV			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Object	{PrimaryBits : [], SecondaryBits: ["h0"]}	{PrimaryBits : [], SecondaryBits: ["'h0"]}	{PrimaryBits : [6], SecondaryBits : ["'h0"]}
Constraint/Dependency	Max 2 12-way interleaving functions. Primary bit defines the granularity of the interleaving. SecondaryBits are not user visible. PrimaryBits array length is 1 and takes an integer between 6 and 12.			
Customer Description	Twelve-way interleaving function. Selects the granularity of the interleaving. - Min value is 6 (64B granularity) - Max value is 12 (4kB) There is a maximum of two such functions.			
Engineering Description	Twelve-way interleaving function to be used inside Ncore Address map. User visible name in the GUI should be "interleaving Granularity". If we ever support hashing this would need to instead allow the customer to set an array of primary and secondary bits			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.8	User-GUI	
Change History	New for 3.8			

TABLE 6-17: DMIINTERLEAVEINFO.DMIINTRLVFUNCTIONS.SIXTEENWAYINTRLVFUNCV

Parameter Name	EightWayIntrlvFuncV			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Object	{PrimaryBits : [], SecondaryBits: ["h0","h0","h0","'h0"]}	{PrimaryBits : [], SecondaryBits: ["'h0","h0","'h0","'h0"]}	{PrimaryBits : [6,7,8,9], SecondaryBits : ["'h0","'h0","h0","'h0"]}
Constraint/Dependency	Max 2 16-way interleaving functions. SecondaryBits are not user visible. PrimaryBits array length is 4 and takes integer between 6 and wAddr-1			
Customer Description	Selects the bit used for the sixteen-way memory interleaving function. It takes four bits between 6 and address width -1. There is a maximum of two such functions and the bit used should not overlap with the bit used to generate the sets in the caches and the directory.			
Engineering Description	Sixteen-way interleaving function to be sued inside Ncore Address map. "Primary Interleaving Bits (4 bits)". If we ever support hashing we would need to allow the customer to also set the secondary bits.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.8	User-GUI	
Change History	Already present in earlier version but not documented this way			

TABLE 6-18: GIU2WIFV GIU 2 WAY INTERLEAVING FUNCTION

Parameter Name	system.GIUInterleaveInfo.giu2WIFV			
Value	Data Type	Architecture	Release	Default
	object	{PrimaryBits : [], SecondaryBits: ["h0"]}	{PrimaryBits : [], SecondaryBits: ["h0"]}	{PrimaryBits : [6], SecondaryBits: ["h0"]}
Constraint/Dependency	Max 2 2-way interleaving functions. SecondaryBits are not user visible. PrimaryBits array length is 4 and takes integer between 6 and wAddr-1			
Customer Description	Selects the bit used for the two-way GIU interleaving function. It takes four bits between 6 and address width - 1. There is a maximum of two such functions and the bit used should not overlap with the bit used to generate the sets in the caches and the directory.			
Engineering Description	Two-way interleaving function to be sued inside Ncore Address map. "Primary Interleaving Bits (1 bit)". If we ever support hashing we would need to allow the customer to also set the secondary bits.			
Release Info	Status	Effective version	Visibility	
	Preview	3.8	Engineering	
Change History				

TABLE 6-19: GIU3WIFV GIU 3 WAY INTERLEAVING FUNCTION

Parameter Name	system.GIUInterleaveInfo.giu3WIFV			
Value	Data Type	Architecture	Release	Default
	object			
Constraint/Dependency	Max 2 12-way interleaving functions. Primary bit defines the granularity of the interleaving. SecondaryBits are not user visible. PrimaryBits array length is 1 and takes an integer between 6 and 12.			
Customer Description	Three-way interleaving function for GIU. Selects the granularity of the interleaving. - Min value is 6 (64B granularity) - Max value is 12 (4kB) There is a maximum of two such functions.			
Engineering Description	Three-way interleaving function to be used inside Ncore Address map. User visible name in the GUI should be "interleaving Granularity". If we ever support hashing this would need to instead allow the customer to set an array of primary and secondary bits			
Release Info	Status	Effective version	Visibility	
	Preview	3.8	Engineering	
Change History				

TABLE 6-20: GIU2WIFV GIU 4 WAY INTERLEAVING FUNCTION

Parameter Name	system.GIUInterleaveInfo.giu2WIFV			
Value	Data Type	Architecture	Release	Default
	object	{PrimaryBits : [], SecondaryBits: ["h0", "h0"]}	{PrimaryBits : [], SecondaryBits: ["h0", "h0"]}	{PrimaryBits : [6,7], SecondaryBits: ["h0","h0"]}
Constraint/Dependency	Max 2 4-way interleaving functions. SecondaryBits are not user visible. PrimaryBits array length is 2 and takes integer between 6 and wAddr-1			
Customer Description	Selects the bit used for the two-way GIU interleaving function. It takes four bits between 6 and address width - 1. There is a maximum of two such functions and the bit used should not overlap with the bit used to generate the sets in the caches and the directory.			

Engineering Description	Four-way interleaving function to be sued inside Ncore Address map. “Primary Interleaving Bits (2 bits)”. If we ever support hashing we would need to allow the customer to also set the secondary bits.		
Release Info	**Status**	**Effective version**	**Visibility**
	Preview	3.8	Engineering
Change History			

7. Socket User Settable Parameters

7.1. Native interface parameters

TABLE 7-1: FNNATIVEINTERFACE PARAMETER

Parameter Name	fnNativeInterface			
Value	Data Type	Architecture	Release	Default
	Valid Values	ACE, ACE5, ACE-LITE, ACE5-LITE, AXI4, AXI5, CHI-B, CHI-E	ACE, ACE5,ACE-LITE, ACE5-LITE, AXI4, AXI5, CHI-B, CHI-E	CHI B
Constraint/Dependency				
Customer Description	Selects native interface type for a CAIU			
Engineering Description	Selects native interface type AXI* results in NCAIU with base modules of IOAIU ACE* results in CAIU with base module of IOAIU ACE*_Lite results in NCAIU with base module of IOAIU CHI* result in CAIU with base module of CHI AIU			
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History	CHI-A support is deprecated in 3.6			

Field Code Changed

All the interfaces are defined based on AXI_Interface (except APB.) That is, AXI_interface parameters are defined first and each interface parameters will be defined on top of it. Will be overwritten if there is any duplicated parameters.

The parameter in this chapter is only for CDTI (Control and data transport interconnect). For the CSTI (Control and status transport interconnect) , all the parameters are fixed and described in Chapter 26.8.

Limitations:

- Header user bit: wArUser, wAwUser[2]. These values per socket must be all the same and must be the same for all Sockets.
- wArUser = wAwUser for all the sockets in the request and response network.
- All Sockets need to have the same address width.

7.1.1. Smallest Coherent Configurations for Ncore

As a coherent interconnect, a Ncore is expected to have at least one coherent agent in any configuration, this implies that the end user needs to configure at least one agent with native CHI* interface or one agent with ACE interface.

Smallest DVM subsystem (Pending on future customer request)

[2] User bits in W, R and B channels are not listed and supported.

With the introduction of ARM DVM v8.4 support, one smallest coherent configuration needs to be added where a Ncore can have at least two agents with ACE5-LITE interfaces enabled with DVM functions. In this configuration, Ncore can have zero CHI* or ACE agent because ARM **DVM v8.4** functionalities are only supported via ACE5-LITE interface (with DVM enabled)[3]. Also, Bidirectional support is always assumed if DVM_Message_Support is not defined.

7.2. AXI4 Interface

TABLE 7-2: PARAMETER RELATED WITH AXI4 INTERFACE: WARID

Parameter Name	wArID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 20/32	Min: 1 Max: 20/32	6
Constraint/Dependency				
Customer Description	Specify the ArId width of AXI4 interface . Initiator: AIU: [1:20], Target: DMI/DII: [1:28]			
Engineering Description	There is a constraint between initiator and target ArID width. Target ArID width must be equal or larger than (maximum of all the AxIDs and wLPId) + wFUnitId. The maximum size is 28 bits for the current release. Make it maximum as 32 bits to leave some room for future growth.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Maximum wAxID for AIU is increased from 10-bit to 20-bit in 3.6.4			

Field Code Changed

TABLE 7-3: PARAMETER RELATED WITH AXI4 INTERFACE: WAWID

Parameter Name	wAwID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 20/32	Min: 1 Max: 20/32	6
Constraint/Dependency				
Customer Description	Specify the AwId width of AXI4 interface . Initiator: AIU: [1:20], Target: DMI/DII: [1:28]			
Engineering Description	There is a constraint between initiator and target AwID width. Target AwID width must be equal or larger than (maximum of all the AxIDs and wLPId) + wFUnitId. The maximum size is 28 bits for the current release. Make it maximum as 32 bits to leave some room for future growth.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Maximum wAxID for AIU is increased from 10-bit to 20-bit in 3.6.4			

Field Code Changed

Table 7-4: Parameter related with AXI4 Interface: wAddr

[3] Ncore ACE5-LITE interface with DVM enabled complies to ARM ACE-Lite version E back in the development time. Now it is mapped to ACE5-LiteDVM interface defined in AXI protocol version IHI0022H.c (ID012621).

Parameter Name	wAddr			
Value	Data Type	Architecture	Release	Default
	Integer	Min: 12 Max: 64	Min: 12 Max: 64	32
Constraint/Dependency				
Customer Description	Specify the width of AXI4 interface address bits.			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

444 TABLE 7-5: PARAMETER RELATED WITH AXI4 INTERFACE: WDATA

Parameter Name	wData			
Value	Data Type	Architecture	Release	Default
	Valid values	[32', '64', '128', '256'] • AIU - 64/128/256 • DII - 64/128/256 • ConfigDII: 32 • DMI - 128/256	[32', '64', '128', '256'] • AIU - 64/128/256 • DII - 64/128/256 • ConfigDII: 32 • DMI - 128/256	AIU/DII: 64 DMI: 128
Constraint/Dependency				
Customer Description	Specify the width of AXI4 interface data bits. Following limitations apply. AXI4 interface connected to memory as Ncore master(DMI): 128 & 256. AXI interface connected to peripheral device Ncore master (DII): 64, 128 & 256. AXI interface connected to a master agent accelerator, GPU, GIC etc. as Ncore slave (AIU): 64, 128 & 256.			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-6: PARAMETER RELATED WITH AXI4 INTERFACE: AWUSER

Parameter Name	wAwUser			
Value	Data Type	Architecture	Release	Default
	Integer	Min: 0 Max: 32	Min: 0 Max: 32	0
Constraint/Dependency				
Customer Description	Width of user bit on AW AXI4 Interface			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

447

TABLE 7-7: PARAMETER RELATED WITH AXI4 INTERFACE: ARUSER

Parameter Name	wArUser			
Value	Data Type	Architecture	Release	Default

	Integer	Min: 0 Max: 32	Min: 0 Max: 32	0
Constraint/Dependency				
Customer Description	Width of user bit on AR AXI4 Interface			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

450 TABLE 7-8: EXCLUSIVE_ACCESSES PROPERTY FOR AXI4 INTERFACE

Parameter Name	exclusiveAccesses			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency				
Customer Description	Once it is turned on, Exclusive accesses are supported			
Engineering Description	Once it is turned on, Exclusive accesses are supported			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-9: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR AXI4 INTERFACE

Parameter Name	useSysCoInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	False
Constraint/Dependency				
Customer Description				
Engineering Description	Shall be set to False to an AXI4 interface-based agent no matter hasProxyCache parameter defined in Table 16-10 is set as true or false. The SysCo handshake is always carried out via the usage of CSR.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

453

7.3. AXI5 Interface

TABLE 7-10: PARAMETER RELATED WITH AXI5 INTERFACE: WARID

Parameter Name	wArID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 20/32	Min: 1 Max: 20/32	6
Constraint/Dependency				
Customer Description	Specify the ArId width of AXI5 interface . Initiator: AIU: [1:20], Target: DMI/DII: [1:28]			
Engineering Description	There is a constraint between initiator and target ArID width. Target ArID width must be equal or larger than (maximum of all the AxIDs and wLPId) + wFUnitId. The maximum size is 28 bits for the current release. Make it maximum as 32 bits to leave some room for future growth.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History	Maximum wAxID for AIU is increased from 10-bit to 20-bit in 3.6.4			

Field Code Changed

TABLE 7-11: PARAMETER RELATED WITH AXI5 INTERFACE: WAWID

Parameter Name	wAwID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 20/32	Min: 1 Max: 20/32	6
Constraint/Dependency				
Customer Description	Specify the AwId width of AXI5 interface. Initiator: AIU: [1:20], Target: DMI/DII: [1:28]			
Engineering Description	There is a constraint between initiator and target AwID width. Target AwID width must be equal or larger than (maximum of all the AxIDs and wLPId) + wFUnitId. The maximum size is 28 bits for the current release. Make it maximum as 32 bits to leave some room for future growth.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History	Maximum wAxID for AIU is increased from 10-bit to 20-bit in 3.6.4			

Field Code Changed

TABLE 7-12: PARAMETER RELATED WITH AXI5 INTERFACE: WADDR

Parameter Name	wAddr			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 12 Max: 64	Min: 12 Max: 64	32
Constraint/Dependency				
Customer Description	Specify the width of AXI5 interface address bits.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-13: PARAMETER RELATED WITH AXI5 INTERFACE: WDATA

Parameter Name	wData			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid values	[32', '64', '128', '256']	[32', '64', '128', '256']	AIU/DII: 64, DMI: 128
Constraint/Dependency	32-bit is only for SysDII			
Customer Description	Specify the width of AXI5 interface data bits. Following limitations apply. AXI4 interface connected to memory as Ncore master(DMI): 128 & 256. AXI interface connected to peripheral device Ncore master (DII): 64, 128 & 256. AXI interface connected to a master agent accelerator, GPU, GIC etc. as Ncore slave (AIU): 64, 128 & 256.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-14: PARAMETER RELATED WITH AXI5 INTERFACE: AWUSER

Parameter Name	wAwUser			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 32	Min: 0 Max: 32	0
Constraint/Dependency				
Customer Description	Width of user bit on AW AXI5 Interface			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-15: PARAMETER RELATED WITH AXI5 INTERFACE: ARUSER

Parameter Name	wArUser			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 32	Min: 0 Max: 32	0
Constraint/Dependency				
Customer Description	Width of user bit on AR AXI5 Interface			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-16: EXCLUSIVE_ACCESSES PROPERTY FOR AXI5 INTERFACE

Parameter Name	exclusiveAccesses			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True

Constraint/Dependency			
Customer Description	Once it is turned on, Exclusive accesses are supported		
Engineering Description			
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.7	User-GUI
Change History			

Field Code Changed

TABLE 7-17: CHECK_TYPE PROPERTY FOR AXI5 INTERFACE

Parameter Name	checkType			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	*NONE, ODD_PARITY_BYTE_ALL*	*NONE, ODD_PARITY_BYTE_ALL*	NONE
Constraint/Dependency				
Customer Description	Odd parity checking included for all signals. Each bit of the parity signal generally covers up to 8 bits.			
Engineering Description	When checkType property is enabled, extra signals need to be added to the AXI5 interface across all channels. Please refer to Table E2-2 of AMBA AXI and ACE Protocol Specification (version ARM IHI 0022H.c ID012621) for the complete signal list.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-18: ATOMIC_TRANSACTIONS PROPERTY FOR AXI5 INTERFACE

Parameter Name	atomicTransactions			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	False
Constraint/Dependency	This property can only be configured as True when proxy cache is disable.			
Customer Description	Once this property is enabled, atomic transactions are supported for the AXI5 interface			
Engineering Description	When atomicTransactions property is enabled, extra relevant signals need to be added to the AXI5 interface across all channels. Please refer to AMBA AXI and ACE Protocol Specification (version ARM IHI 0022H.c ID012621) for the complete signal list.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-19: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR AXI5 INTERFACE

Parameter Name	useSysCoInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	False
Constraint/Dependency	Can only be configured as true when hasProxyCahce parameter defined in Table 16-10 is set as true.			
Customer Description	It is a user settable feature and can be set as true if hasProxyCache parameter defined in Table 16-10 is set as true.			

Engineering Description	Default False to an AXI5 interface-based CPU no matter hasProxyCahce parameter defined in Table 16-10 is set as true or false. It is a user settable feature and can be set as true if hasProxyCache parameter defined in Table 16-10 is set as true.		
Release Info	**Status**	**Effective version**	**Visibility**
	Experimental	3.7	Engineering
Change History			

Field Code Changed

7.4. ACE Interface

TABLE 7-20: PARAMETER RELATED WITH ACE INTERFACE: WARID

Parameter Name	wArID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 20	Min: 1 Max: 20	6
Constraint/Dependency				
Customer Description	Specify the ArId width of ACE interface .			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-21: PARAMETER RELATED WITH ACE INTERFACE: WAWID

Parameter Name	wAwID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 20	Min: 1 Max: 20	6
Constraint/Dependency				
Customer Description	Specify the AwId width of ACE interface .			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-22: PARAMETER RELATED WITH ACE INTERFACE: WADDR

Parameter Name	wAddr			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	32, 40, 44, 48	32, 40, 44, 48	32
Constraint/Dependency				
Customer Description	Specify the width of ACE interface address bits.			
Engineering Description	ACE only: 32, 40, 44, 48, ACE with CHI: 44, 48			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-23: PARAMETER RELATED WITH ACE INTERFACE: WDATA

Parameter Name	wData			
Value	Data Type	Architecture	Release	Default
	Valid values	64, 128, 256	64, 128, 256	64
Constraint/Dependency				
Customer Description	Specify the width of ACE interface data bits			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-24: PARAMETER RELATED WITH ACE INTERFACE: AWUSER

Parameter Name	wAwUser			
Value	Data Type	Architecture	Release	Default
	Integer	Min: 0 Max: 32	Min: 0 Max: 32	0
Constraint/Dependency				
Customer Description	Width of AwUser bit on AW channel of ACE Interface			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-25: PARAMETER RELATED WITH ACE INTERFACE: ARUSER

Parameter Name	wArUser			
Value	Data Type	Architecture	Release	Default
	Integer	Min: 0 Max: 32	Min: 0 Max: 32	0
Constraint/Dependency				
Customer Description	Width of ArUser bit on AR channel of ACE Interface			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-26: EXCLUSIVE_ACCESSES PROPERTY FOR ACE INTERFACE

Parameter Name	exclusiveAccesses			
Value	Data Type	Architecture	Release	Default
	Boolean	True, False	True, False	True
Constraint/Dependency				
Customer Description	Once it is turned on, Exclusive accesses are supported			

Engineering Description	Once it is turned on, Exclusive accesses are supported		
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	User-GUI
Change History			

Field Code Changed

TABLE 7-27: PARAMETER RELATED TO EVENTIN INTERFACE

Parameter Name	useEventInInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency				
Customer Description	Setting the parameter to enable the connection of EventInReq and EventInAck interface to the AIU.			
Engineering Description	Connect the I/O to the SysReq Receiver hardware. Default True to ACE interface CPU, but if customer's CPU does not have the interface, it can be set as False, or tie signal EventInAck to EventInReq			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-28: PARAMETER RELATED TO EVENTOUT INTERFACE

Parameter Name	useEventOutInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency				
Customer Description	Setting the parameter to enable the connection of EventOutReq and EventOutAck interface to the AIU.			
Engineering Description	Connect the I/O to the SysReq Sender hardware. Default True to ACE interface CPU, but if customer's CPU does not have the interface, it can be set as False, or tie signal EventOutReq to 1'b0			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-29: SHAREABLE_TRANSACTIONS PROPERTY FOR ACE INTERFACE

Parameter Name	shareableTransactions			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency				
Customer Description	Once it is turned on, AxDOMAIN signals will present in ACE interface, inner and outer shareable transactions are supported			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-30: CONTINUOUS_CACHE_LINE_READ_DATA PROPERTY FOR ACE INTERFACE

Parameter Name	continuousCacheLineReadData			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency				
Customer Description	Once it is turned on, continuous cache line data return is supported for the ACE interface			
Engineering Description	Once it is turned on, continuous cache line data return is supported for the ACE interface			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-31: DVM_V8 PROPERTY FOR ACE INTERFACE

Parameter Name	dvmV8			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency				
Customer Description	Once it is turned on, DVM version 8.0 functionalities are supported for the ACE interface			
Engineering Description	Once it is turned on, DVM version 8.0 functionalities are supported for the ACE interface			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

510

TABLE 7-32: DVM_V8.1 PROPERTY FOR ACE INTERFACE

Parameter Name	dvmV8.1			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency	Once version 8.1 is supported, DVM version 8.0 is also supported and DVM_v8 is a don't care value.			
Customer Description	Once it is turned on, DVM version 8.1 functionalities are supported for the ACE interface.			
Engineering Description	Once it is turned on, DVM version 8.1 functionalities are supported for the ACE interface.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

513 TABLE 7-33: DVM_MESSAGE_SUPPORT PROPERTY FOR ACE INTERFACE

Parameter Name	dvmMessageSupport			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid values	Bidirectional, Receiver, False	Bidirectional, Receiver, False	Bidirectional
Constraint/Dependency				
Customer Description	Bidirectional is always supported in Ncore so this property is fixed and the customer can not change it			
Engineering Description	Bidirectional is always supported in Ncore so this property is fixed and the customer can not change it			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 7-34: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR ACE INTERFACE

Parameter Name	useSysCoInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	False
Constraint/Dependency				
Customer Description	Shall be set to False to an ACE interface-based agent. The SysCo handshake is always carried out via the usage of CSR.			
Engineering Description	Shall be set to False to an ACE interface-based agent. The SysCo handshake is always carried out via the usage of CSR.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	No	
Change History				

Field Code Changed

7.5. ACE5 Interface

519

TABLE 7-35: PARAMETER RELATED WITH ACE5 INTERFACE: WARID

Parameter Name	wArID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 20	Min: 1 Max: 20	6
Constraint/Dependency				
Customer Description	Specify the ArId width of ACE5 interface .			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

522 TABLE 7-36: PARAMETER RELATED WITH ACE5 INTERFACE: WAWID

Parameter Name	wAwID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 20	Min: 1 Max: 20	6
Constraint/Dependency				
Customer Description	Specify the AwId width of ACE5 interface.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-37: PARAMETER RELATED WITH ACE5 INTERFACE: WADDR

Parameter Name	wAddr			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	32, 40, 44, 48	32, 40, 44, 48	32
Constraint/Dependency				
Customer Description	Specify the width of ACE5 interface address bits.			
Engineering Description	ACE only: 32, 40, 44, 48 ACE with CHI: 44, 48			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

525

TABLE 7-38: PARAMETER RELATED WITH ACE5 INTERFACE: WDATA

Parameter Name	wData			
Value	**Data Type**	**Architecture**	**Release**	**Default**

	Valid values	64, 128, 256	64, 128, 256	64
Constraint/Dependency				
Customer Description	Specify the width of ACE5 interface data bits			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

528 TABLE 7-39: PARAMETER RELATED WITH ACE5 INTERFACE: AWUSER

Parameter Name	wAwUser			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 32	Min: 0 Max: 32	0
Constraint/Dependency				
Customer Description	Width of AwUser bit on AW channel of ACE5 Interface			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-40: PARAMETER RELATED WITH ACE5 INTERFACE: ARUSER

Parameter Name	wArUser			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 32	Min: 0 Max: 32	0
Constraint/Dependency				
Customer Description	Width of ArUser bit on AR channel of ACE5 Interface			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

531

TABLE 7-41: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR ACE5 INTERFACE

Parameter Name	useSysCoInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	False
Constraint/Dependency				
Customer Description	Setting the parameter to enable or disable the connection of SysCoReq and SysCoAck interface to the IOAIU. if customer's CPU does not support SysCo interface, and does not set False to the parameter, it is recommended tying SysCoReq to 0 and the handshake has to be carried out via the usage of CSR.			
Engineering Description	Connect the I/O to the SysCo Engine hardware. Default false to an ACE5 interface-based CPU. If customer's CPU does have the SysCo interface, customers can set the parameter to be true.			
Release Info	**Status**	**Effective version**	**Visibility**	

	Active	3.7	User-GUI
Change History			

Field Code Changed

TABLE 7-42: PARAMETER RELATED TO EVENTIN INTERFACE

Parameter Name	useEventInInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency				
Customer Description	Setting the parameter to enable the connection of EventInReq and EventInAck interface to the AIU.			
Engineering Description	Connect the I/O to the SysReq Receiver hardware. Default True to ACE interface CPU, but if customer's CPU does not have the EventIn interface, customers can set the parameter to be False or to tie EventInAck to EventInReq.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-43: PARAMETER RELATED TO EVENTOUT INTERFACE

Parameter Name	useEventOutInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency				
Customer Description	Setting the parameter to enable the connection of EventOutReq and EventOutAck interface to the AIU.			
Engineering Description	Default True to ACE5 interface CPU, but if customer's CPU does not have the EventOut interface, customers can set the parameter to be False or to tie EventOutReq to 0.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-44: CHECK_TYPE PROPERTY FOR ACE5 INTERFACE

Parameter Name	checkType			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	*NONE, ODD_PARITY_BYTE_ALL*	*NONE, ODD_PARITY_BYTE_ALL*	NONE
Constraint/Dependency				
Customer Description	Odd parity checking included for all signals. Each bit of the parity signal generally covers up to 8 bits.			
Engineering Description	Once checkType property is enabled, extra signals need to be added to the AXI5 interface across all channels. Please refer to Table E2-2 of AMBA AXI and ACE Protocol Specification (version ARM IHI 0022H.c ID012621) for the complete signal list.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-45: EXCLUSIVE_ACCESSES PROPERTY FOR ACE5 INTERFACE

Parameter Name	exclusiveAccesses

Value	Data Type	Architecture	Release	Default
	Boolean	True, False	True, False	True
Constraint/Dependency				
Customer Description	Once it is turned on, Exclusive accesses are supported			
Engineering Description	Once this property is enabled, extra relevant signals need to be added to the ACE5 interface across all channels. Please refer to AMBA AXI and ACE Protocol Specification (version ARM IHI 0022H.c ID012621) for the complete signal list			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-46: SHAREABLE_TRANSACTIONS PROPERTY FOR ACE5 INTERFACE

Parameter Name	shareableTransactions			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency				
Customer Description	Once it is turned on, AxDOMAIN signals will present in ACE5 interface, inner and outer shareable transactions are supported			
Engineering Description	Once it is turned on, AxDOMAIN signals will present in ACE5 interface, inner and outer shareable transactions are supported			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-47: CONTINUOUS_CACHE_LINE_READ_DATA PROPERTY FOR ACE5 INTERFACE

Parameter Name	continuousCacheLineReadData			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency				
Customer Description	Once it is turned on, continuous cache line data return is supported for the ACE5 interface			
Engineering Description	Once it is turned on, continuous cache line data return is supported for the ACE5 interface			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

TABLE 7-48: DVM_V8 PROPERTY FOR ACE5 INTERFACE

Parameter Name	dvmV8			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency				
Customer Description	Once it is turned on, DVM version 8.0 functionalities are supported for the ACE5 interface			
Engineering Description	Once it is turned on, DVM version 8.0 functionalities are supported for the ACE5 interface			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	

Field Code Changed

Change History	

TABLE 7-49: DVM_V8.1 PROPERTY FOR ACE5 INTERFACE

Parameter Name	dvmV8			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency	Once version 8.1 is supported, DVM version 8.0 is also supported and DVM_v8 is a don't care value.			
Customer Description	Once it is turned on, DVM version 8.1 functionalities are supported for the ACE5 interface.			
Engineering Description	Once it is turned on, DVM version 8.1 functionalities are supported for the ACE5 interface.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	User-GUI	
Change History				

Field Code Changed

549

TABLE 7-50: DVM_MESSAGE_SUPPORT PROPERTY FOR ACE5 INTERFACE

Parameter Name	dvmMessageSupport			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid values	Bidirectional, Receiver, False	Bidirectional, Receiver, False	Bidirectional
Constraint/Dependency				
Customer Description	Currently, bidirectional is always supported in Ncore so this property is fixed and the customer can not change it			
Engineering Description	Currently, bidirectional is always supported in Ncore so this property is fixed and the customer can not change it			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7	Engineering	
Change History				

Field Code Changed

552

7.6. ACE5-LITE Interface

555

TABLE 7-51: PARAMETER RELATED WITH ACE5-LITE/ACE5-LITEDVM INTERFACE: WARID

Parameter Name	wArID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 20	Min: 1 Max: 20	6
Constraint/Dependency				
Customer Description	Specify the ArID width of ACE5-Lite/ACE5-LiteDVM interface.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

558 TABLE 7-52: PARAMETER RELATED WITH ACE5-LITE/ACE5-LITEDVM INTERFACE: WAWID

Parameter Name	wAwID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 20	Min: 1 Max: 20	6
Constraint/Dependency				
Customer Description	Specify the width of ACE5-Lite/ACE5-LiteDVM interface AwId bits.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-53: PARAMETER RELATED WITH ACE5-LITE/ACE5-LITEDVM INTERFACE: WADDR

Parameter Name	wAddr			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 12 Max: 64	Min: 12 Max: 64	32
Constraint/Dependency				
Customer Description	Specify the width of ACE5-Lite/ACE5-LiteDVM interface Address bits.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

561

TABLE 7-54: PARAMETER RELATED WITH ACE5-LITE/ACE5-LITEDVM INTERFACE: WDATA

Parameter Name	wData			
Value	**Data Type**	**Architecture**	**Release**	**Default**

	Valid values	64, 128, 256	64, 128, 256	64
Constraint/Dependency				
Customer Description	Specify the width of ACE5-Lite/ACE5-LiteDVM interface data bits			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

564

TABLE 7-55: PARAMETER RELATED WITH ACE5-LITE/ACE5-LITEDVM INTERFACE: AWUSER

Parameter Name	wAwUser			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 32	Min: 0 Max: 32	0
Constraint/Dependency				
Customer Description	Width of user bit on AW ACE5-Lite/ACE5-LiteDVM Interface			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

567 TABLE 7-56: PARAMETER RELATED WITH ACE5-LITE/ACE5-LITEDVM INTERFACE: ARUSER

Parameter Name	wArUser			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 32	Min: 0 Max: 32	0
Constraint/Dependency				
Customer Description	Width of user bit on AR ACE5-LITE/ACE5-LiteDVM Interface			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-57: PARAMETER TO ENABLE DVM FUNCTIONALITIES FOR THE ACE5-LITE INTERFACE: DVM

Parameter Name	DVM			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid values	True, False	True, False	False
Constraint/Dependency	Set this parameter false will turn off ANY DVM functionalities			
Customer Description	Set this parameter true will make ACE5-Lite interface to be an ACE5-LiteDVM interface.			
Engineering Description	Once it is set as true, the properties such as DVM_v8, DVM_v8.1, DVM_v8.4, DVM_Message_Support and Coherency_Connection_Signals properties start to take effect. If it is set as FALSE, DVM_v8, DVM_v8.1 and DVM_v8.4, Coherency_Connection_Signals and DVM_Message_Support should NOT be available for selection.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

570

TABLE 7-58: DVM_V8 PROPERTY FOR ACE5-LITEDVM INTERFACE

Parameter Name	dvmV8			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	False
Constraint/Dependency	Only available for selection when DVM parameter is set as TRUE.			
Customer Description	Once it is turned on, DVM version 8.0 functionalities are supported for the ACE5-LiteDVM interface			
Engineering Description	Once it is turned on, DVM version 8.0 functionalities are supported for the ACE5-Lite DVM interface			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-59: DVM_V8.1 PROPERTY FOR ACE5-LITEDVM INTERFACE

Parameter Name	dvmV8.1			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency	Only available for selection when DVM parameter is set as TRUE..			
Customer Description	Once it is turned on, DVM version 8.1 functionalities are supported for the ACE5-LiteDVM interface.			
Engineering Description	Once it is turned on, DVM version 8.1 functionalities are supported for the ACE5-LiteDVM interface. Once version 8.1 is supported, DVM version 8.0 is also supported and DVM_v8 is a don't care value			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-60: DVM_V8.4 PROPERTY FOR ACE5-LITEDVM INTERFACE

Parameter Name	dvmV8.4			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	False
Constraint/Dependency	Only available for selection when DVM parameter is set as TRUE.			
Customer Description	Once it is turned on, DVM version 8.4 functionalities are supported for the ACE5-LiteDVM interface.			
Engineering Description	Once it is turned on, DVM version 8.4 functionalities are supported for the ACE5-LiteDVM interface. Once version 8.4 is supported, DVM version 8.0 and 8.1 are also supported, DVM_v8 and DVM_v8.1 values are don't care values.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-61: DVM_MESSAGE_SUPPORT PROPERTY FOR ACE5-LITEDVM INTERFACE

Parameter Name	dvmMessageSupport			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid values	Bidirectional, Receiver, False	Bidirectional, Receiver, False	Bidirectional
Constraint/Dependency	Only take effect when DVM parameter is set as TRUE.			
Customer Description	Currently, bidirectional is always supported in Ncore so this property is fixed and the customer can not change it			

Engineering Description	Currently, bidirectional is always supported in Ncore so this property is fixed and the customer can not change it		
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	Engineering
Change History			

Field Code Changed

579

TABLE 7-62: PARAMETER RELATED TO EVENTOUT INTERFACE

Parameter Name	useEventOutInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	False
Constraint/Dependency				
Customer Description	Setting the parameter to enable the connection of EventOutReq and EventOutAck interface to the AIU. if customer's CPU does not have the interface, and does not set False to the parameter, it is recommended to tie EventOutReq to 0.			
Engineering Description	Connect the I/O to the SysReq Sender hardware. Default True to ACE5-Lite interface CPU, but if customer's CPU does not have the EventOut interface, customers can set the parameter to be False. It should be set as true once DVM functions are supported (when eAC=1)			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

582 TABLE 7-63: PARAMETER TO ENABLE THE PROTECTION ON THE ACE5-LITE/ACE5-LITEDVM INTERFACE

Parameter Name	checkType			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	*NONE, ODD_PARITY_BYTE_ALL*	*NONE, ODD_PARITY_BYTE_ALL*	NONE
Constraint/Dependency				
Customer Description	Default to not enable the protection on the ACE5- Lite/ACE5-LiteDVM interface. Necessary odd parity signals across ALL channels will be added once it is set as ODD_PARITY_BYTE_ALL			
Engineering Description	Necessary odd parity signals across ALL channels for an ACE5- Lite/ACE5-LiteDVM interface will be added once it is set as ODD_PARITY_BYTE_ALL.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-64: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR ACE5-LITEDVM INTERFACE

Parameter Name	useSysCoInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	False
Constraint/Dependency	This parameter can only be selectable with DVM parameter is enabled, and it will be default as false under such a condition.			
Customer Description	Setting the parameter to enable or disable the connection of SysCoReq and SysCoAck interface to the IOAIU. if customer's CPU does not have the interface, and does not set False to the parameter, it is recommended tying the SysCoReq to 0, and the SysCo handshake is carried out via the usage of CSR.			
Engineering Description	Connect the I/O to the SysCo Engine hardware. Default false to an ACE5-LiteDVM interface-based CPU. If customer's CPU does have the SysCo interface, customers can set the parameter to be true.			

Release Info	Status	Effective version	Visibility
	Active	3.2	User-GUI
Change History			

Field Code Changed

TABLE 7-65: ATOMIC_TRANSACTIONS PROPERTY FOR ACE5-LITE/ACE5-LITEDVM INTERFACE

Parameter Name	atomicTransactions			
Value	Data Type	Architecture	Release	Default
	Boolean	True, False	True, False	False
Constraint/Dependency	This property can only be selectable if the proxy cache is disable.			
Customer Description	Once this property is enabled, atomic transactions are supported for the ACE5-Lite/ACE5-LiteDVM interface			
Engineering Description	Once this property is enabled, atomic transactions are supported for the ACE5-Lite/ACE5-LiteDVM interface			
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-66: EXCLUSIVE_ACCESSES PROPERTY FOR ACE5-LITE/ACE5-LITEDVM INTERFACE

Parameter Name	exclusiveAccesses			
Value	Data Type	Architecture	Release	Default
	Boolean	True, False	True, False	True
Constraint/Dependency				
Customer Description	Once it is turned on, Exclusive accesses are supported			
Engineering Description	Once it is turned on, Exclusive accesses are supported			
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

591 TABLE 7-67: SHAREABLE_TRANSACTIONS PROPERTY FOR ACE5-LITE/ACE5-LITEDVM INTERFACE

Parameter Name	shareableTransactions			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency				
Customer Description	Once it is turned on, AxDOMAIN signals will present in ACE5-Lite/ACE5-LiteDVM interface, inner and outer shareable transactions are supported			
Engineering Description	Once it is turned on, AxDOMAIN signals will present in ACE5-Lite/ACE5-LiteDVM interface, inner and outer shareable transactions are supported			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-68: CACHE_STASH_TRANSACTIONS PROPERTY FOR ACE5-LITE INTERFACE

Parameter Name	CacheStashTransactions			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency	Always set as true for ACE5_Lite interface			
Customer Description				
Engineering Description	Once it is turned on, cache stash transactions are supported for the ACE5-Lite interface. Please note that this property is NOT expected to be enabled and used with PCAIU configuration enabled.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

594

TABLE 7-69: TRACE_SIGNALS PROPERTY FOR ACE5-LITE INTERFACE

Parameter Name	eTrace			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency	Always set as true for ACE5_Lite interface			
Customer Description				
Engineering Description	Once it is turned on, the relevant *TRACE signales will be added across different channels for the ACE5-Lite interface. Please note this property is also be turned on for ACE5_LiteDVM interface once the DVM functionalities are enbled via what it is defined in **Error! Reference source not found.**.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

597

7.7. ACE-LITE Interface

TABLE 7-70: PARAMETER RELATED WITH ACE-LITE INTERFACE: WARID

Parameter Name	wArID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 20	Min: 1 Max: 20	6
Constraint/Dependency				
Customer Description	Specify the ArID width of ACE-LITE interface.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-71: PARAMETER RELATED WITH ACE-LITE INTERFACE: WAWID

Parameter Name	wAwID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 20	Min: 1 Max: 20	6
Constraint/Dependency				
Customer Description	Specify the AwId width of ACE-LITE interface.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-72: PARAMETER RELATED WITH ACE-LITE INTERFACE: WADDR

Parameter Name	wAddr			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 12 Max: 64	Min: 12 Max: 64	32
Constraint/Dependency				
Customer Description	Specify the address width of ACE-LITE interface .			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-73: PARAMETER RELATED WITH ACE-LITE INTERFACE: WDATA

Parameter Name	wData			
Value	**Data Type**	**Architecture**	**Release**	**Default**

	Valid values	64, 128, 256	64, 128, 256	64
Constraint/Dependency				
Customer Description	Specify the data width of ACE-LITE interface			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

609

Table 7-74: Parameter related with ACE-LITE Interface: AwUser

Parameter Name	wAwUser			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 32	Min: 0 Max: 32	0
Constraint/Dependency				
Customer Description	Width of AwUser bits on AW channel of ACE-LITE Interface			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

612 TABLE 7-75: PARAMETER RELATED WITH ACE-LITE INTERFACE: ARUSER

Parameter Name	wArUser			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 32	Min: 0 Max: 32	0
Constraint/Dependency				
Customer Description				
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-76: PARAMETER RELATED WITH ACE-LITE INTERFACE: EAC

Parameter Name	eAC			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid values	0, 1	0, 1	0
Constraint/Dependency				
Customer Description	Not applicable for ACE-Lite interface, DVM functions can be enabled by ACE5-Lite interface			
Engineering Description	Architecture team would remove the parameter in the future NCore versions.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Deprecated	3.2	Engineering	
Change History				

Field Code Changed

TABLE 7-77: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR ACE-LITE INTERFACE

Parameter Name	useSysCoInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	False
Constraint/Dependency				
Customer Description				
Engineering Description	Shall be set to False to an ACE-Lite interface-based agent. The SysCo handshake is always done via the usage of CSR.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

Table 7-78: Shareable_Transactions Property for ACE-Lite Interface

Parameter Name	shareableTransactions			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency				
Customer Description	Once it is turned on, AxDOMAIN signals will present in ACE-Lite interface, inner and outer shareable transactions are supported			
Engineering Description	Once it is turned on, AxDOMAIN signals will present in ACE-Lite interface, inner and outer shareable transactions are supported			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 7-79: EXCLUSIVE_ACCESSES PROPERTY FOR ACE-LITE INTERFACE

Parameter Name	exclusiveAccesses			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	True
Constraint/Dependency				
Customer Description	Once it is turned on, Exclusive accesses are supported			
Engineering Description	Once it is turned on, Exclusive accesses are supported			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

7.8. CHI Issue B Interface

624

TABLE 7-80: PARAMETER RELATED WITH CHI_B_INTERFACE: NODEID_WIDTH

Parameter Name	NodeID_Width			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		Min: 7 Max: 11	Min: 7 Max: 11	7
Constraint/Dependency				
Customer Description	Width of the Node ID of the CHI Interface.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

627 TABLE 7-81: PARAMETER RELATED WITH CHI_B_INTERFACE: WADDR

Parameter Name	wAddr			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 44 Max: 52	Min: 44 Max: 52	48
Constraint/Dependency				
Customer Description	Width of the address on CHI interface.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-82: PARAMETER RELATED WITH CHI_B_INTERFACE: REQ_RSVDC

Parameter Name	REQ_RSVDC			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	['0', '4', '8','12', '16', '24', '32']	['0', '4', '8','12', '16', '24', '32']	0
Constraint/Dependency				
Customer Description	REQ_RSVDC is to define user-bit for command channel. Do not support user bit on data channel.			
Engineering Description				
Release Info	***Status***	***Effective version***	***Visibility***	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

630

TABLE 7-83: PARAMETER RELATED WITH CHI_B_INTERFACE: WDATA

Parameter Name	wData			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	128, 256	128, 256	128

Constraint/Dependency			
Customer Description	Width of data on the Chi interface		
Engineering Description			
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	User-GUI
Change History			

Field Code Changed

TABLE 7-84: PARAMETER RELATED WITH CHI_B_INTERFACE: ENPOISON

Parameter Name	enPoison			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	True, False	True, False	False
Constraint/Dependency				
Customer Description	Enable Poison Bit			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-85: PARAMETER RELATED TO SYSCO INTERFACE

Parameter Name	useSysCoInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency				
Customer Description	Setting the parameter to enable the connection of SysCoReq and SysCoAck interface to the AIU.			
Engineering Description	Connect the I/O to the SysCo Engine hardware. Default True to CHI-B interface CPU, but if customer's CPU does not have the SysCo interface, customers can set the parameter to be False or tie SysCoReq to 0			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-86: PARAMETER RELATED TO EVENTIN INTERFACE

Parameter Name	useEventInInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency				
Customer Description	Setting the parameter to enable the connection of EventInReq and EventInAck interface to the AIU.			
Engineering Description	Connect the I/O to the SysReq Receiver hardware. Default True to CHI-B interface CPU, if customer's CPU does not have the interface, this parameter can be set to false or to tie EventInAck to EventInReq.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-87: PARAMETER RELATED TO EVENTOUT INTERFACE

Parameter Name	useEventOutInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency				
Customer Description	Setting the parameter to enable the connection of EventOutReq and EventOutAck interface to the AIU.			
Engineering Description	Connect the I/O to the SysReq Sender hardware. Default True to CHI-B interface CPU, but if customer's CPU does not have the EventOut interface, customers can set the parameter to be False or to tie EventOutReq to 0.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 7-88: PARAMETER TO ENABLE THE PROTECTION ON THE CHI-B INTERFACE

Parameter Name	checkType			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	*NONE, ODD_PARITY_BYTE_ALL*	*NONE, ODD_PARITY_BYTE_ALL*	NONE
Constraint/Dependency	Default to not enable the protection on the CHI-B interface. Necessary odd parity signals across ALL channels will be added once it is set as ODD_PARITY_BYTE_ALL			
Customer Description	Enable signal protection on CHI-B interface			
Engineering Description	Necessary odd parity signals across ALL channels will be added once it is set as ODD_PARITY_BYTE_ALL.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

7.9. CHI Issue E Interface

TABLE 7-89: PARAMETER RELATED WITH CHI_E_INTERFACE: NODEID_WIDTH

Parameter Name	NodeID_Width			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		Min: 7 Max: 11	Min: 7 Max: 11	7
Constraint/Dependency				
Customer Description	Width of the Node ID of the CHI Interface.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.6	User-GUI	
Change History				

Field Code Changed

TABLE 7-90: PARAMETER RELATED WITH CHI_E_INTERFACE: WADDR

Parameter Name	wAddr			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 44 Max: 52	Min: 44 Max: 52	48
Constraint/Dependency				
Customer Description	Width of the address on CHI interface.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.6	User-GUI	
Change History				

Field Code Changed

TABLE 7-91: PARAMETER RELATED WITH CHI_E_INTERFACE: REQ_RSVDC

Parameter Name	REQ_RSVDC			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	['0', '4', '8','12', '16', '24', '32']	['0', '4', '8','12', '16', '24', '32']	0
Constraint/Dependency				
Customer Description	REQ_RSVDC is to define user-bit for command channel. Do not support user bit on data channel.			
Engineering Description				
Release Info	***Status***	***Effective version***	***Visibility***	
	Active	3.6	User-GUI	
Change History				

Field Code Changed

TABLE 7-92: PARAMETER RELATED WITH CHI_E_INTERFACE: WDATA

Parameter Name	wData			
Value	**Data Type**	**Architecture**	**Release**	**Default**

	Valid Values	128, 256	128, 256	128
Constraint/Dependency				
Customer Description	Width of data on the Chi interface			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.6	User-GUI	
Change History				

Field Code Changed

Table 7-93: Parameter related with CHI_E_Interface: enPoison

Parameter Name	**enPoison**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	True, False	True, False	False
Constraint/Dependency				
Customer Description	Enable Poison Bit			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.6	User-GUI	
Change History				

Field Code Changed

TABLE 7-94: PARAMETER RELATED TO SYSCO INTERFACE

Parameter Name	useSysCoInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency				
Customer Description	Setting the parameter to enable the connection of SysCoReq and SysCoAck interface to the AIU.			
Engineering Description	Connect the I/O to the SysCo Engine hardware. Default True to CHI-E interface CPU, but if customer's CPU does not have the SysCo interface, customers can set the parameter to be False or to tie SysCoReq to 0.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.6	User-GUI	
Change History				

Field Code Changed

TABLE 7-95: PARAMETER RELATED TO EVENTIN INTERFACE

Parameter Name	useEventInInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency				
Customer Description	Setting the parameter to enable the connection of EventInReq and EventInAck interface to the AIU.			

Engineering Description	Connect the I/O to the SysReq Receiver hardware. Default True to CHI-E interface CPU, but if customer's CPU does not have the EventIn interface, customers can set the parameter to be False or to tie EventInAck to EventInReq.		
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.6	User-GUI
Change History			

Field Code Changed

Table 7-96: parameter related to eventout interface

Parameter Name	useEventOutInt			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency				
Customer Description	Setting the parameter to enable the connection of EventOutReq and EventOutAck interface to the AIU.			
Engineering Description	Connect the I/O to the SysReq Sender hardware. Default True to CHI-E interface CPU, but if customer's CPU does not have the EventOut interface, customers can set the parameter to be False or tie EventOutReq to 0.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.6	User-GUI	
Change History				

Field Code Changed

TABLE 7-97: PARAMETER TO ENABLE THE PROTECTION ON THE CHI-E INTERFACE

Parameter Name	checkType			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	*NONE, ODD_PARITY_BYTE_ALL*	*NONE, ODD_PARITY_BYTE_ALL*	NONE
Constraint/Dependency	Default to not enable the protection on the CHI-E interface. Necessary odd parity signals across ALL channels will be added once it is set as ODD_PARITY_BYTE_ALL			
Customer Description	Enable signal protection for CHI-E interface			
Engineering Description	Necessary odd parity signals across ALL channels will be added once it is set as ODD_PARITY_BYTE_ALL.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.6	User-GUI	
Change History				

Field Code Changed

7.10. CXS.B Interface

669

TABLE 7-98: PARAMETER TO SELECT THE NUMBER OF CXS.B LINK CREDITS

Parameter Name	CXS_MAX_CREDIT			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	*4-15*	*4-15*	15
Constraint/Dependency				
Customer Description	This parameter sets the maximum number of link credit of the CXS interfaces (**CXS_MAX_CREDIT**). It applies to both the TX and RX ports.			
Engineering Description	This parameter sets the maximum number of link credit of the CXS interfaces (**CXS_MAX_CREDIT**). It applies to both the TX and RX ports.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Preview	3.8	User-GUI	
Change History				

Field Code Changed

672 TABLE 7-99: PARAMETER TO SELECT CXS.B CHECK_TYPE

Parameter Name	CXSCHECKTYPE			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	*None, Odd_Byte_Parity*	*None, Odd_Byte_Parity*	None
Constraint/Dependency				
Customer Description	This parameter controls the protection of the CXS interface (**CXSCHECKTYPE**).			
Engineering Description	This parameter controls the protection of the CXS interface(**CXSCHECKTYPE**).			
Release Info	**Status**	**Effective version**	**Visibility**	
	Preview	3.8	User-GUI	
Change History				

Field Code Changed

8. *Concerto User Settable Parameters*

All Concerto parameters are derived parameters.

9. CAIU User Settable Parameters

9.1. CAIU resource parameters

TABLE 9-1: NOTTCTRLENTRIES FOR CAIU

Parameter Name	nOttCtrlEntries			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 8 Max: 128	Min: 8 Max: 128	8
Constraint/Dependency				
Customer Description	Specify the maximum number of outstanding native transactions this AIU should support.			
Engineering Description	An AIU converts certain inbound native agent requests into protocol coherent transactions and allocate resources in the AIU Outstanding Transaction Table (OTT). This parameter configures the size of OTT table			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 9-2: GENERIC PORTS PARAMETER FOR CAIU

Parameter Name	genericports			
Value	**Data Type**	**Architecture**	**Release**	**Default**
Constraint/Dependency				
Customer Description				
Engineering Description	To assign user defined ports for place holder definition(Resiliency); Described in Chapter 27.1 Applied to CHI-AIU and IOAIU			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 9-3: MEMORY PARAMETER FOR CAIU

Parameter Name	Memory			
Value	**Data Type**	**Architecture**	**Release**	**Default**
Constraint/Dependency				
Customer Description				
Engineering Description	This parameter is to assign SRAM. For the memory setting, refer Chapter 24.2 This is only for ACE.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

9.2. CAIU credit parameters

690

TABLE 9-4: NNATIVECREDITS PARAMETER FOR CAIU

Parameter Name	nNativeCredits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 2 Max: 15	Min: 2 Max: 15	2
Constraint/Dependency				
Customer Description	Specify the maximum number of CHI link credits this AIU should support.			
Engineering Description	The number of credits will be specified for each flow, they define how many transactions can be in flight between an initiator and a target.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

693 TABLE 9-5: NSTASHSNPCREDITS FOR CAIU

Parameter Name	nStashSnpCredits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 8	Min: 1 Max: 8	2
Constraint/Dependency				
Customer Description	Specify the maximum number of outstanding stash snoops this AIU should support. These are stash snoops issued on the CHI interface.			
Engineering Description	This is used for assign Ott Stash entries in CAIU. Total number of OTT entries = nOttCtrlEntries + nStshSnpCredits			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

696 TABLE 9-6: NDCECMDCREDITS FOR CAIU

Parameter Name	nDceCmdCredits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 2 Max: 16	Min: 2 Max: 16	2
Constraint/Dependency				
Customer Description	Specify the maximum number of credits for coherent transactions per DCE. This should be determined based on required bandwidth and network round trip latency.			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-Register	
Change History	Credit is software programmable since 3.4 (Visibility: User-GUI → User-Register)			

Field Code Changed

699

TABLE 9-7: NDMICMDCREDITS FOR CAIU

Parameter Name	nDmiCmdCredits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 2 Max: 16	Min: 2 Max: 16	2
Constraint/Dependency	Applied to CHI-AIU and IOAIU			
Customer Description	Specify the maximum number of credits for non-coherent transactions per DMI. This should be determined based on required bandwidth and network round trip latency.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-Register	
Change History	Credit is software programmable since 3.4 (Visibility: User-GUI → User-Register)			

Field Code Changed

702

705

TABLE 9-8: NDIICMDCREDITS FOR CAIU

Parameter Name	nDiiCmdCredits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 2 Max: 16	Min: 2 Max: 16	2
Constraint/Dependency				
Customer Description	Specify the maximum number of credits for non-coherent transactions per DII. This should be determined based on required bandwidth and network round trip latency.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-Register	
Change History	Credit is software programmable since 3.4 (Visibility: User-GUI → User-Register)			

Field Code Changed

708

711

9.3. CAIU address map parameter

TABLE 9-9: FNCSRACCESS_PARAMETER

Parameter Name	fnCsrAccess			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	True, False	True, False	True
Constraint/Dependency	Should be true on at-least one AIU, cannot be true for AXI AIU with NcMode as false. Applied to CHI-AIU and IOAIU			

Customer Description	Enable CSR access via this AIU		
Engineering Description	Parameter works as a reset value for CSR BAR valid bit.		
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	User-GUI
Change History			

Field Code Changed

9.4. CAIU snoop filter parameters

TABLE 9-10: SNOOPFILTER_REF PARAMETER FOR CAIUS

Parameter Name	SnoopFilter_Ref			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 64	Min: 0 Max: 64	0
Constraint/Dependency				
Customer Description	Specify the snoop filter associated with corresponding CAIU			
Engineering Description	User would need to bind CAIU into specific snoop filter, using update_object -name $caiu_name -type snoopFilter -bind $ snoop_filter_name. Archi team would want to move this parameter to DCE in next NCore versions.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

9.5. CAIU performance counter parameters

TABLE 9-11: CAIU PERFORMANCE COUNTER PARAMETERS

Parameter Name	nPerfCounters (CAIU/IOAIU)			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 16	Min: 0 Max: 8	4
Constraint/Dependency	Only three valid values are supported. 0, 4 and 8. Applied to CHI-AIU and IOAIU.			
Customer Description	Total number of performance counter in NCore Unit. Each counter can be configured to count different events present in an Ncore unit via CSRs, please refer to the reference manual on the details of performance counter event.			
Engineering Description	Once the nPerfCounters is configured with a value bigger than zero, there will be 16 Latency Counters connected automatically by the hardware. And if the nPerfCounters is configured with a value of zero, NO Latency counters will be connected by the hardware, which implies the performance monitoring feature is completely disabled for this CAIU. The nPerfCounters value of 16 triggers an implementation issue and therefore is not an option for CAIU.			
Release Info	Status	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 9-12: CAIU LATENCY COUNTER PARAMETERS

Parameter Name	nLatencyCounters			
Value	Data Type	Architecture	Release	Default
	Integer	Min: 0 Max: 32	Min: 0 Max: 16	16
Constraint/Dependency				
Customer Description	Number of Latency counters in a CAIU.			
Engineering Description	Only two valid values are supported 0 or 16. A non-zero value is possible only if nPerfCounters is greater than or equal to 4.			
Release Info	Status	Effective version	Visibility	
	Active	3.2	Engineering	
Change History				

Field Code Changed

9.6. CAIU processor info parameters

Ncore supports exclusive monitors by creating a basic monitor for each core in each DCE and a configurable number of tagged monitors in each DCE.

Each basic monitor implements the behavior described in the "Minimum PoS Exclusive Monitor" section in the ACE specification, and each tagged monitor implements the behavior described in the "Additional address comparison" section.

The number of cores performing an exclusive sequence **MUST** be specified per CAIU **(nProcessors)**. In the case of ACE-CAIU the ARID and AWID bits that identify the core performing an exclusive access sequence must be specified **(AxIdProcSelectBits).**

TABLE 9-13: NPROCESSOR PARAMETERS FOR CAIU

Parameter Name	nProcessors			
Value	Data Type	Architecture	Releas	
	Enum	CHI-CAIU: 1, 2, 4, 8, 16 IOAIU: 1, 2, 4, 8, 16, 32	CHI-CAIU: 1, 2, 4, 8, 16 IOAIU: 1, 2, 4, 8, 16, 32	1
Constraint/Dependency				
Customer Description	Number of Processors			
Engineering Description	CHI-AIU: SW must multiply the specified parameter by 2 before passing it on to RTL. This is to account for threads as each core can have up to two threads. For the ACE, we should not do this shift.			
Release Info	Status	Effective version	Visibility	
	Deprecated	3.2	User-GUI	
Change History	Deprecated in 3.8. Updated to nExclusiveThreads.			

Field Code Changed

747 TABLE 9-14: NEXCLUSIVETHREADS PARAMETERS FOR CAIU

Parameter Name	nExclusiveThreads			
Value	Data Type	Architecture	Release	
	Integer	No requirement	1-32	1
Constraint/Dependency				
Customer Description	Max number of exclusive threads that this agent can generate. An exclusive thread is defined as an exclusive read/write sequence. It is typically related to the number of logical processors connected to the interface.			
Engineering Description	Max number of exclusive thread that this agent can generate. This is used to size the exclusive monitor size in DCE. This is equivalent to the nProcessor parameter in earlier version except it is never multiplied by 2. The customer becomes responsible to account for multiple thread if needed. It replaces nProcessors, when migrating it should take the value originally passed to RTL i.e multiply nProcessor by 2 for CHI and use the same number for an ACE.			
Release Info	Status	Effective version	Visibility	
	Preview	3.8	User-GUI	
Change History				

Field Code Changed

TABLE 9-15: AXIDPROCSELECTBITS PARAMETERS FOR CAIU

Parameter Name	AxIdProcSelectBits			
Value	***Data Type***	***Integer Array***	***Integer Array***	
	Array			
Constraint/Dependency	Only applies for ACE. Array must contain number smaller than AXID. The length of the array must match log2ceil(nExclusiveThread)			
Customer Description	Processor Select Bits from AXID. If there is only one processor, the array is empty and is default as zero			
Engineering Description	This is used to generate an index for exclusive thread coming from an ACE AIU to be used inside DCE			
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

750

9.7. CAIU SysCmd Hardware parameters

753 The following parameters are used to instantiate specific hardware within the CAIU to process sysco/event messages. The following parameters should be visible to Engineering team only.

756 TABLE 9-16: USESYSCOENGINE PARAMETERS FOR CAIU

Parameter Name	useSysCoEngine			
Value	Data Type	Architecture	Release	Default
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency				
Customer Description				
Engineering Description	Used to instantiate SysCo Engine hardware in the AIU Always True for ACE/CHI/AXI with Proxy Cache AIUs. if useSysCoInt is True, set True to this parameter			
Release Info	Status	Effective version	Visibility	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 9-17: USESYSREQSENDER PARAMETERS FOR CAIU

Parameter Name	useSysReqSender			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency				
Customer Description				
Engineering Description	Used to instantiate SysReq Sender hardware in the AIU Always True for ACE/CHI AIUs, optional for ACE_Lite + DVM AIUs . if useEventOutInt is True, set True to this parameter			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 9-18: USESYSREQRECEIVER PARAMETERS FOR CAIU

Parameter Name	useSysReqReceiver			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency				
Customer Description				
Engineering Description	Used to instantiate SysReq Receiver hardware in the AIU Always True for ACE/CHI AIUs , if useEventInInt is True, set True to this parameter			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

9.8. CAIU Connectivity parameters

The following parameters are used to specify connectivity information of the CAIU. The following parameters should be visible to the Engineering team only.

TABLE 9-19: HEXAIUDCEVEC PARAMETERS FOR CAIU

Parameter Name	hexAiuDceVec			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: FFFFFFFF	Min: 0 Max: FFFF	1
Constraint/Dependency	Size of the vector is equal to the number of DCEs in the system. Every bit in the vector that is set to one represents a DCE at that NodeID that is connected to the AIU.			
Customer Description				
Engineering Description	This must be a port in RTL (tACHL) and tie off parameter in SW Every bit in the vector that is set to one specifies that the particular AIU is connected to the associated DCE at that NunitID			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	

Field Code Changed

Change History	

Table 9-20: hexaiudmivec parameters for CAIU

Parameter Name	hexAiuDmiVec			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: FFFFFFFF	Min: 0 Max: FFFF	*1*
Constraint/Dependency	Size of the vector is equal to the number of DMIs in the system. Every bit in the vector that is set to one represents a DMI at that NodeID that is connected to the AIU.			
Customer Description				
Engineering Description	This must be a port in RTL (tACHL) and tie off parameter in SW Every bit in the vector that is set to one specifies that the particular AIU is connected to the associated DMI at that NunitID			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

771

TABLE 9-21: HEXAIUDIIVEC PARAMETERS FOR CAIU

Parameter Name	hexAiuDiiVec			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: FFFFFFFF	Min: 1 Max: FFFF	1
Constraint/Dependency	Size of the vector is equal to the number of DIIs in the system. Every bit in the vector that is set to one represents a DII at that NodeID that is connected to the AIU. By default we must have one DII, which is sysDII, to be configured in an Ncore.			
Customer Description				
Engineering Description	This must be a port in RTL (tACHL) and tie off parameter in SW Every bit in the vector that is set to one specifies that the particular AIU is connected to the associated DII at that NunitID			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

774 TABLE 9-22: HEXAIUCONNECTEDDCEFUNITID PARAMETERS FOR CAIU

Parameter Name	hexAiuConnectedDceFunitId			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: FFFFFFFF	Min: 0 Max: FFFF	1
Constraint/Dependency	List of DCE Funit IDs that are connected to the AIU. This list can be ordered in Nunit ID order.			
Customer Description				
Engineering Description	This must be a port in RTL (tACHL) and tie off parameter in SW. List of DCE FuntilDs that are connected to the AIU.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	

Field Code Changed

Change History	

Table 9-23: nAiuConnectedDces parameters for CAIU

Parameter Name	nAiuConnectedDces			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 64	Min: 1 Max: 16	1
Constraint/Dependency	Number of DCEs connected to this AIU.			
Customer Description				
Engineering Description	Specifies the number of DCEs that are connected to this AIU			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

10. *CAIU non-User settable parameters*

TABLE 10-1: NSTTCTRLENTRIES FOR CAIU

Parameter Name	nSttCtrlEntries			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Derived	Derived	Derived
Constraint/Dependency	$\sum_{nDCEs} nAiuSnpCredits + nDvmSnpCredits + (nDies - 1) \times nRemoteSnpCredits$			
Customer Description	Specifies the maximum number of outstanding snoop transactions the AIU can receive.			
Engineering Description	Specifies the maximum number of outstanding snoop transactions the AIU can receive.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Preview	3.8	Engineering	
Change History				

Field Code Changed

11. *DCE User Settable Parameters*

11.1. DCE resource parameters

TABLE 11-1: NATTCTRLENTRIES PARAMETER

Parameter Name	nAttCtrlEnries			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 4 Max: 96	Min: 4 Max: 96	4
Constraint/Dependency				
Customer Description	Specify the maximum number of active coherent transactions tracked by each DCE.			
Engineering Description	When DCE accepts a CMDreq for processing, it allocates an entry in the ATT (Active Transaction Table) where it stores all persistent fields from a message. The entry will remain allocated until the transaction has completed in the system. The number of entries needs to be determined by analyzing performance requirements (BW, latency) and configuring the ATT size for each DCE.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Maximum value is increased from 64 to 96 in v3.7			

Field Code Changed

TABLE 11-2: NCMDSKIDBUFSIZE PARAMETER

Parameter Name	nCMDSkidBufSize			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 4 Max: ~~768~~8448	Min: 4 Max: 8448~~768~~	16
Constraint/Dependency	restrict granularity, supporting only sizes: nCMDSkidBufArb + {0, 4, 8, 16, 32, 64, 96, 128, 160, 192, 224, 256,512, 1024,2048, 4096, 8192}			
Customer Description	Total depth of skid buffer for commands to DCE/DII and the non-coherent port of DMI. The skid buffer is used to stage transaction requests from initiator agents. The number of required entries may be determined by traffic requirements and analysis using performance modeling. This value sets the total budget of protocol credits available for distribution. CSR Address: 0xFF0			
Engineering Description	This value sets the total budget of protocol credits available for distribution to communicating initiators. It is recommended to allow at least 2 credits for each active connection. For a specific DCE inside of a Ncore, the maximum value of nCMDSkidBufSize should be the total sum of nDceCmdCredits from any connected CAIU(s) and nDceCmdCredits from any connected NCAIU(s). During the configuration, It is recommended to do a sanity check to the value of nCMDSkidBufSize for any DCE with a formula of (2*(the number of connected CAIUs + the number of connected NCAIUs))[4], which assuming the minimum nDceCmdCredits of 2 from each connected agent.			
Release Info	**Status**	**Effective version**	**Visibility**	

[4] This formula does NOT guarantee the DCE works to the end user's specification but only serves the purpose of flagging any misconfiguration of this parameter

	Active	3.2	User-GUI
Change History	Maximum value is increased from 512 to 768 in v3.7 Max increased to 8448 in 3.8		

Field Code Changed

TABLE 11-3: NCMDSKIDBUFARB PARAMETER

Parameter Name	**nCMDSkidBufArb**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 4 Max: 256	Min: 4 Max: 256	16
Constraint/Dependency	≤ nCMDSkidBufSize restricted granularity, support only sizes: 4, 8, 16, 32, 48, 64, 128, 192, 256			
Customer Description	Depth of skid buffer visible to arbitration. This value determines the size of the arbitration window within which arriving requests are selected based on QoS, priority and arrival time.			
Engineering Description	This value sets the number of entries within the skid buffer that is visible to arbitration. CSR Address: 0xFF0 It is recommended to start with a reasonably value for performance analysis - the area of a skid buffer grows with the square of this number and larger options will also significantly impact timing.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Maximum value is support up to 256 from 64 in v3.6			

Field Code Changed

11.2. DCE credit parameters

TABLE 11-4: NDCERBCREDITS PARAMETER

Parameter Name	**nDceRbCredits**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 2 Max: 64	Min: 2 Max: 64	2
Constraint/Dependency				
Customer Description	Number of RB credits per DCE			
Engineering Description	The value is same for all DCEs and DMIs Specify the maximum number of DCE write request buffer credits per DMI. These credits limit number of Coherent writes and incudes snoops that can cause a write to DMI.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Maximum value is increased from 32 to 64 in v3.7			

Field Code Changed

TABLE 11-5: NAIUSNPCREDITS PARAMETER

Parameter Name	**nAiuSnpCredits**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 2 Max: 16	Min: 2 Max: 16	2
Constraint/Dependency				
Customer Description	Specify the maximum number of snoop request credits per AIU.			

Engineering Description			
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	User-GUI
Change History			

Field Code Changed

TABLE 11-6: NDMIMRDCREDITS PARAMETER

Parameter Name	**nDmiMrdCredits**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 2 Max: 16	Min: 2 Max: 16	2
Constraint/Dependency				
Customer Description	Specify the maximum number of memory read credits per DMI.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-Register	
Change History	Credit is software programmable since 3.4 (Visibility: User-GUI → User-Register)			

Field Code Changed

11.3. DCE snoop filter parameters

TABLE 11-7: NSETS SF CONFIGURATION PARAMETERS

Commented [BM4]: Review

Parameter Name	**nSets**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 16 Max: 128K	Min: 16 Max: 128K	16
Constraint/Dependency				
Customer Description	Number of sets for the selected snoop filter			
Engineering Description	Number of sets must be divisible by number of DCEs in the system.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Maximum sets is increased to 128K in v3.7			

Field Code Changed

TABLE 11-8: NWAYS SF CONFIGURATION PARAMETERS

Parameter Name	**nWays**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 2 Max: 32	Min: 2 Max: 32	4
Constraint/Dependency				
Customer Description	Number of ways for the selected snoop filter			
Engineering Description				

Release Info	Status	Effective version	Visibility
	Active	3.2	User-GUI
Change History			

Field Code Changed

TABLE 11-9: NVICTIMENTRIES SF CONFIGURATION PARAMETERS

Parameter Name	nVictimEntries			
Value	Data Type	Architecture	Release	Default
	Integer	Min: 0 Max: 64	Min: 0 Max: 64	2
Constraint/Dependency				
Customer Description	Number of victim buffer entries for the specified snoop filter, per DCE			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 11-10: SETSELECTPRIMARYBITV PARAMETERS

Parameter Name	SetSelectPrimaryBitV			
Value	Data Type	Architecture	Release	
	Array of Integers			
Constraint/Dependency				
Customer Description	Set selection parameter.			
Engineering Description	Bits that select the set. They need to be as many as log2(number of sets / number of DCEs), they cannot be in the LSBs inside a cache line, and they cannot overlap with DCE interleaving bits. For example, for a system with 64B cache lines, 1024 sets and 4 DCEs interleaved on bits [11 : 10], this needs to be an 8-bit array with values that could be e.g. [15, 14, 13, 12, 9, 8, 7, 6].			
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

Table 11-11: Memory parameter for DCE Snoop Filter

Parameter Name	Memory			
Value	Data Type	Architecture	Release	Default
Constraint/Dependency				
Customer Description				
Engineering Description	This parameter is to assign SRAM. For the memory setting, refer to Chapter 24.2			
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

819 TABLE 11-12: REMOTECACHINGAGENTS PARAMETER

Parameter Name	RemoteCachingAgents			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	array of strings			
Constraint/Dependency	Available in DCE, and only valid if associated caching agents is connected to the DCE			
Customer Description	Specify if the caching agents is considered remote to the DCE. Eg. [AIU0, AIU2]			
Engineering Description	Derive the value into array of integers for DCE, to indicate if the corresponding caching agent is remote (sync up with DCE's JSON file parameter)			
Release Info	Status	Effective version	Visibility	
	Preview	3.8	User-GUI	
Change History				

Field Code Changed

822

TABLE 11-12: LOCALCACHINGAGENTS PARAMETERS

Commented [BM5]: Quesaco?

Parameter Name	LocalCachingAgents			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	array of strings			
Constraint/Dependency	Available in DCE, and only valid if associated caching agents is connected to the DCE			
Customer Description	Specify if the caching agents is considered local to the DCE, Eg. [AIU1]			
Engineering Description	Derive the value into array of integers for DCE, to indicate if the corresponding caching agent is local (sync up with DCE's JSON file parameter)			
Release Info	**Status**	**Effective version**	**Visibility**	
	Preview	3.8	User-GUI	
Change History				

Field Code Changed

825

11.4. DCE performance counter parameters

TABLE 11-13: DCE NPERFCOUNTERS PARAMETERS

Parameter Name	nPerfCounters			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 16	Min: 0 Max: 16	0
Constraint/Dependency	Valid values supported are: 0, 4, 8 and 16			
Customer Description	Total number of performance counter in Ncore Unit			
Engineering Description	Architecture team would modify this as a common parameter in next Ncore versions.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

Please note that latency counters are **NOT** used for a DCE. Therefore, there isn't any latency counter is connected no matter the number of performance counter is set as zero, four or eight. This means that the latency counters are only configurable in AIUs, DMIs and DIIs.

11.5. DCE exclusive monitor parameters

An Ncore supports exclusive monitors (only for shareable domain) by creating a basic monitor for each core in each DCE and a configurable number of tagged monitors in each DCE. Each basic monitor implements the behavior described in the "Minimum PoS Exclusive Monitor" section in the ACE specification, and each tagged monitor implements the behavior described in the "Additional address comparison" section.

TABLE 11-14: NTAGGEDMONITORS PARAMETER

Parameter Name	nTaggedMonitors			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 8	Min: 0 Max: 8	0
Constraint/Dependency				
Customer Description	Specify the desired number of tagged exclusive monitor per DCE instance. Note that each DCE instance will always have a basic exclusive monitor.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

11.6. DCE Connectivity parameters

The following parameters are used to specify connectivity information of the DCE. The following parameters should be visible to the engineering team only.

TABLE 11-15: HEXDCECONNECTEDDMIFUNITID PARAMETER

Parameter Name	hexDceConnectedDmiFunitID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: FFFFFFFF	Min: 0 Max: FFFF	1
Constraint/Dependency				
Customer Description				
Engineering Description	List of DMI Funit IDs that are connected to the DCE. This is ordered in Nunit ID order , skipping DMIs not connected to the DCE. This must be a port in RTL (tACHL) and tie off parameter in SW			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 11-16: HEXDCECONNECTEDCAFUNITID PARAMETER

Parameter Name	hexDceConnectedCaFunitID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: FFFFFFFF	Min: 0 Max: FFFF	1
Constraint/Dependency				
Customer Description				
Engineering Description	List of caching agent Funit IDs that are connected to the DCE. This list can be ordered in either snoop filter order or Nunit ID order This must be a port in RTL (tACHL) and tie off parameter in SW			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 11-17: HEXDCEDMIVEC PARAMETER

Parameter Name	hexDceDmiVec			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: FFFFFFFF	Min: 0 Max: FFFF	1
Constraint/Dependency				

Customer Description			
Engineering Description	Size of the vector is equal to the number of DMIs in the system Every bit in the vector that is set to one specifies that the particular DCE is connected to the associated DMI at that NunitID This must be a port in RTL (tACHL) and tie off parameter in SW		
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	Engineering
Change History			

Field Code Changed

TABLE 11-18: HEXDCEDMIRBOFFSET PARAMETER

Parameter Name	hexDceDmiRbOffset			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 32 DMIs	Min: 0 Max: 16 DMIs	1
Constraint/Dependency	The max length (number of bits) is defined as number of DMIs connected to a DCE in the system multiplied by 8 Each 8-bit value represents the DMI connected to that DCE. They are ordered in the increasing order NunitID, skipping DMIs not connected to the DCE. The 8-bit offset value is calculated as follows For every DMI create a vector of all DCEs in the system. Every bit in the vector that is set to one represents a DCE at that NodeID that is connected to the DMI. For the first valid DCE in the vector the offset value is nDceRbCredits * 0 For the second valid DCE in the vector the offset value is nDceRbCredits * 1 So on and so forth This breaks down to a formula as nDceRbCredits * (Dce position - 1)			
Customer Description				
Engineering Description	This must be a port in RTL (tACHL) and tie off parameter in SW List of 8 bit values, where every 8 bit value specifies the RBID offset to be used by DCE for the DMI represented by the value. The offsets are ordered in incrementing DMI NunitID order.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

858

TABLE 11-19: NDCECONNECTEDCAS PARAMETER

Parameter Name	nDceConnectedCas			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 64	Min: 1 Max: 32	1
Constraint/Dependency				
Customer Description				
Engineering Description	Specifies the number of caching agents (CAIUs) that are connected to DCE This parameter must be same for all DCEs			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

861 TABLE 11-20: NDCECONNECTEDDMIS PARAMETER

Parameter Name	nDceConnectedDmis			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 32	Min: 1 Max: 16	1
Constraint/Dependency				

Customer Description			
Engineering Description	Specifies the number of DMIs that are connected to DCE This parameter must be same for all DCEs		
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	Engineering
Change History			

Field Code Changed

TABLE 11-21: NDCERBCREDITS PARAMETER

Parameter Name	nDceRbCredits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 2 Max: 64	Min: 2 Max: 64	2
Constraint/Dependency				
Customer Description				
Engineering Description	Number of RB credits per DCE The value is same for all DCEs and DMIs			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

12. *DCE Non-user Settable Parameters*

TABLE 12-1: NLPIDPERCAAGENT FOR DCE

Parameter Name	nLpIdPerCaAgent			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Derived	Derived	Derived
Constraint/Dependency	**Singel Die** : $\max_{CAIU} nExclusiveThreads$ **Multi Die** : GlobalLargestNExclusiveThreads (from system parameters)			
Customer Description	NA			
Engineering Description	Specifies the largest number of exclusive threads across all CAIUs.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Preview	3.8	Engineering	
Change History				

Field Code Changed

TABLE 12-2: NEXCLUSIVETHREAD FOR DCE

Parameter Name	nExclusiveThread			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Derived	Derived	Derived
Constraint/Dependency	**Single Die** : (nProxyCaches + nIOAIUp + nCAius) * **nLpIdPerCaAgent** **Multi Die :** nGlobalCachingAgents * GlobalLargestNExclusiveThreads (from system parameters)			
Customer Description	NA			
Engineering Description	This is the largest number of exclusive threads across CAIUs times the total number of caching agent. This means that it includes proxy cache and IOAUp which are not capable of sending coherent exclusive.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Preview	3.8	Engineering	
Change History				

Field Code Changed

13. *DMI User Settable Parameters*

13.1. DMI resource parameters

TABLE 13-1: GENERICPORTS PARAMETERS FOR DMI

Parameter Name	genericports			
Value	Data Type	Architecture	Release	Default
Constraint/Dependency				
Customer Description				
Engineering Description	To assign user defined ports for place holder definition (Resiliency); Described in Chapter 27.1			
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 13-2: NRTTCTRLENTRY PARAMETERS

Parameter Name	nRttCtrlEnries			
Value	Data Type	Architecture	Release	Default
	Integer	Min: 4 Max: 128	Min: 4 Max: 128	4
Constraint/Dependency				
Customer Description	Specify the number of allowed outstanding read transactions on the downstream AXI interface.			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 13-3: NWTTCTRLENTRY PARAMETERS

Parameter Name	nWttCtrlEnries			
Value	Data Type	Architecture	Release	Default
	Integer	Min: 4 Max: 64	Min: 4 Max: 64	4
Constraint/Dependency				
Customer Description	Specify number of allowed outstanding write transactions on the downstream AXI interface.			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 13-4: NDMIRBCREDITS PARAMETER

Parameter Name	nDmiRbCredits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 2 Max: 64	Min: 2 Max: 64	2
Constraint/Dependency				
Customer Description	Specify the maximum number of non-coherent write request buffer credits.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Maximum value is increased from 32 to 64 in v3.7			

Field Code Changed

Table 13-5: nCMDSkidBufSize parameter

Parameter Name	nCMDSkidBufSize			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 4 Max: 8448~~768~~	Min: 4 Max: 8448~~768~~	16
Constraint/Dependency	Restrict granularity, supporting only sizes: nCMDSkidBufArb + {0, 4, 8, 16, 32, 64, 96, 128, 160, 192, 224, 256,512, 1024,2048, 4096, 8192}			
Customer Description	Total depth of skid buffer for commands to DCE/DII and the non-coherent port of DMI. The skid buffer is used to stage transaction requests from initiator agents. The number of required entries may be determined by traffic requirements and analysis using performance modeling. This value sets the total budget of protocol credits available for distribution.			
Engineering Description	This value sets the total budget of protocol credits available for distribution to communicating initiators. It is recommended to allow at least 2 credits for each active connection. CSR Address: 0xFF0 For a specific DMI inside of a Ncore, the maximum value of nCMDSkidBufSize should be the total sum of nDmiCmdCredits as defined in Table 9 7 from any connected CAIU(s) and nDmiCmdCredits as defined in Table 14 6 from any connected NCAIU(s). During the configuration, It is recommended to do a sanity check to the value of nCMDSkidBufSize for any DMI with a formula of (2*(the number of connected CAIUs + the number of connected NCAIUs)) , which assuming the minimum nDmiCmdCredits of 2 from each connected agent.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Maximum value is increased from 512 to 768 in v3.7 Max increased to 8448 in 3.8			

Formatted: Font: (Default) Arial

Field Code Changed

TABLE 13-6: NCMDSKIDBUFARB PARAMETER

Parameter Name	nCMDSkidBufArb			
Value	**Data Type**	**Architecture**	**Release**	**Default**

	Integer	Min: 4 Max: 256	Min: 4 Max: 256	16
Constraint/Dependency	≤ nCMDSkidBufSize restricted granularity, support only sizes: 4, 8, 16, 32, 48, 64, 128, 192, 256			
Customer Description	Depth of skid buffer visible to arbitration. This value determines the size of the arbitration window within which arriving requests are selected based on QoS, priority and arrival time. It is recommended to start with a reasonably value for performance analysis - the area of a skid buffer grows with the square of this number and larger options will also significantly impact timing.			
Engineering Description	This value sets the number of entries within the skid buffer that is visible to arbitration CSR Address: 0xFF0			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Maximum value is support up to 256 from 64 in v3.6			

Field Code Changed

TABLE 13-7: NMRDSKIDBUFSIZE PARAMETER

Parameter Name	**nMrdSkidBufSize**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 4 Max: 768	Min: 4 Max: 768	16
Constraint/Dependency	Restrict granularity, supporting only sizes: nMrdSkidBufArb + {0, 4, 8, 16, 32, 64, 96, 128, 160, 192, 224, 256, 512}			
Customer Description	Total depth of skid buffer for coherent DMI transactions - arriving from DCE. The skid buffer is used to stage transaction requests from initiator agents. The number of required entries may be determined by traffic requirements and analysis using performance modeling. This value sets the total budget of protocol credits available for distribution.			
Engineering Description	This value sets the total budget of protocol credits available for distribution to communicating initiators. It is recommended to allow at least 2 credits for each active connection. CSR Address: 0xFE0 For a specific DMI inside of an Ncore, the maximum value of nMrdSkidBufSize should be the total sum of nDmiMrdCredits from any connected DCE(s). During the configuration, It is recommended to do a sanity check to the value of nDmiMrdCredits for any DMI with a formula of (2*(the number of connected DCEs))., which assuming the minimum nDmiMrdCredits of 2 from each connected DCE.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Maximum value is increased from 512 to 768 in v3.7			

Field Code Changed

TABLE 13-8: NMRDSKIDBUFARB PARAMETER

Parameter Name	**nMrdSkidBufArb**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 4 Max: 256	Min: 4 Max: 256	16
Constraint/Dependency	≤ nMrdSkidBufArb restricted granularity, support only sizes: 4, 8, 16, 32, 48, 64, 128, 192, 256			

Customer Description	Depth of skid buffer visible to arbitration. This value determines the size of the arbitration window within which arriving requests are selected based on QoS, priority and arrival time. It is recommended to start with a reasonably value for performance analysis - the area of a skid buffer grows with the square of this number and larger options will also significantly impact timing.		
Engineering Description	This value sets the number of entries within the skid buffer that is visible to arbitration CSR Address: 0xFE0		
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	User-GUI
Change History			

Field Code Changed

TABLE 13-9: ENABLEREADRSPINTRLV PARAMETER

Parameter Name	**enableReadRspIntrlv**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		True, False	True, False	False
Constraint/Dependency				
Customer Description	Use this parameter to enable the feature of DMI can accept read data interleaving from AXI interface			
Engineering Description	To prevent deadlock issue of AXI write address channel, write response channel and read data channel, if the parameter is set to True, and read data buffer is instantiated. And DMI can accept any beat of read data of issued read request, then the read data/response channel and write response channel will never to backpressured.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	User-GUI	
Change History				

Field Code Changed

TABLE 13-10: NEXCLUSIVEENTRIES PARAMETER

Parameter Name	**nExclusiveEntries**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 8	Min: 0 Max: 8	0
Constraint/Dependency				
Customer Description	defines the number of exclusive monitors			
Engineering Description	A value of 0 means no exclusive monitor will be instantiated			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.6	User-GUI	
Change History				

Field Code Changed

13.2. DMI address map parameters

906 TABLE 13-11: NADDRTRANSREGISTERS PARAMETERS

Parameter Name	nAddrTransRegisters			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 16	Min: 0 Max: 16	0
Constraint/Dependency				
Customer Description	Specifies the number of address translation registers that are available within DMI. These registers add capability to translate address on the AXI bus from DMI. Refer to the address translation section of the reference manual.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Changed to 16 on 9/17/2025			

Field Code Changed

909 TABLE 13-12: DIRECTMEMORYMODE PARAMETERS

Parameter Name	DirectMemoryMode			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True/false	True/False	True
Constraint/Dependency	Only exist if a non-power of two interleaving functions have been defined			
Customer Description	This option should be selected if it is required by the target to have a contiguous address range with no holes.due to interleaving with a non-power of two. Typically this option should be selected when DMI connects directly to a memory controller and can be disabled when it connects into another non-coherent interconnect which will become responsible to deinterleave the traffic.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.8	User-GUI	
Change History				

13.3. DMI System Cache parameters

912

TABLE 13-13: DMI SYSTEM CACHE ENABLE PARAMETERS

Parameter Name	useCachee			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	True, False	True, False	False
Constraint/Dependency				
Customer Description	This option adds an SMC in DMI. It must be enabled when an atomic capable master is present in the system and requires at least a 4KB SMC.			
Engineering Description				

Release Info	Status	Effective version	Visibility
	Active	3.2	User-GUI
Change History			

Field Code Changed

TABLE 13-14: DMI SCRATCHPAD ENABLE PARAMETERS

Parameter Name	useScratchPad			
Value	***Data Type***	***Valid Values***	***Valid values***	
	Boolean	True, False	True, False	False
Constraint/Dependency	Can be enabled only when system cache is enabled.			
Customer Description	Enable ScratchPad			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

Table 13-15: DMI Cache Way Partitioning Registers parameters

Parameter Name	nWayPartitioningRegisters			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 16	Min: 0 Max: 16	0
Constraint/Dependency				
Customer Description	Specifies the number of cache way partitioning registers. Each register enables configuration capability to assign specific ways to a single agent. The number of registers enabled here should be equal to maximum number of agents that will be configured for way partitioning.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 13-16: DMI CACHE NTAGBANK CONFIGURATION PARAMETERS

Parameter Name	nTagBanks			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 4	Min: 1 Max: 4	1
Constraint/Dependency	Values limited to 1, 2,4.			
Customer Description	Number of Tag banks.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Maximum Tag banks is increased from 2 to 4 in v3.7			

Field Code Changed

TABLE 13-17: DMI CACHE NDATABANK CONFIGURATION PARAMETERS

Parameter Name	nDataBanks			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 4	Min: 1 Max: 4	1
Constraint/Dependency	Values limited to 1, 2.4			
Customer Description	Number of Data banks.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 13-18: MEMORY PARAMETER SMC

Parameter Name	Memory			
Value	**Data Type**	**Architecture**	**Release**	**Default**

Constraint/Dependency			
Customer Description			
Engineering Description	This parameter is to assign SRAM. For the memory setting, refer to Chapter 24.2 for detail		
Release Info	**Status**	**Effective version**	**Visibility**
	Active	*3.2*	*User-GUI*
Change History			

Field Code Changed

13.4. DMI performance counter parameters

TABLE 13-19: DMI NPERFCOUNTERS PARAMETERS

Parameter Name	nPerfCounters			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 16	Min: 0 Max: 16	4
Constraint/Dependency	Valid values supported are: 0, 4, 8 and 16			
Customer Description	Total number of performance counter in a DMI.			
Engineering Description	Architecture team would modify this as a common parameter in next Ncore versions.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

Once the nPerfCounters is configured with a value bigger than zero, there will be 16 Latency Counters connected automatically by the hardware. And if the nPerfCounters is configured with a value of zero, NO Latency counters will be connected by the hardware, which implies the performance monitoring feature is completely disabled for this DMI.

TABLE 13-20: DMI LATENCY COUNTER PARAMETERS

Parameter Name	nLatencyCounters			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 32	Min: 0 Max: 16	16
Constraint/Dependency	Only two valid values are supported 0 or 16. A non-zero value is possible only if nPerfCounters is greater than or equal to 4.			
Customer Description	Number of Latency counters in a DMI.			
Engineering Description	Parameter applies only to AIUs, DMIs and DIIs only and can be set individlually.			
Release Info	Status	Effective version	Visibility	
	Active	3.2	Engineering	
Change History				

Field Code Changed

13.5. DMI Atomic parameters

The SMC offers an option to include an Atomic Engine (AE). The AE supports the Far Atomic Operations (FAOs) defined in CHI-B and ACE5-LITE interface architectures. Thus, in Ncore 3 FAOs are supported for all locations in system memory connected via the DMI

TABLE 13-21: USEATOMIC PARAMETERS

Parameter Name	useAtomic			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid values	True/false	True/false	false
Constraint/Dependency	Can only be set to true if one of the interface supports atomics			
Customer Description	Set to true to enable the atomic engine. It can only be set to true if a system cache is also present.			
Engineering Description	Set to true to enable the atomic engine. It can only be set to true if a system cache is also present.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

945

Field Code Changed

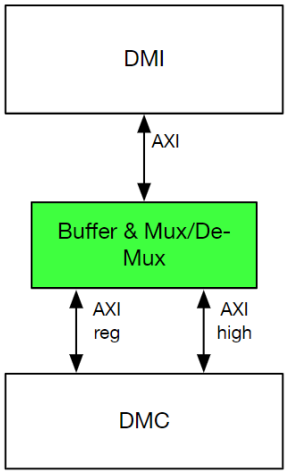
13.6. DMI transaction table reservation for improved QoS

948 The customer/user of Ncore is expected to develop the following assumptions apply in this use case

The DMC used has 2 AXI ports one for regular traffic shown as “AXI reg” and another for high priority or real time traffic shown as “AXI high”

951 The user or customer develops “Buffer & Mux/De-Mux”block

The “Buffer & mux/de-mux” block consists of simple logic where it has a buffer that is larger than the DMC’s AXI reg port buffer. The mux/de-mux logic is responsible for routing the high priority or real time traffic to DMC’s AXI high, while all other
954 traffic is routed to DMC’s AXI reg port. The buffer being larger than the buffer DMC’s AXI reg port buffer grantees that high priority traffic does not see head of line blocking.



957 FIGURE 2 PROPOSED ARCHITECTURE TO TAKE ADVANTAGE OF TT RESERVATIONS

TABLE 13-22: DMIQOSTHVAL PARAMETERS

Parameter Name	DmiQoSThVal			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		Min: 1 Max: 15	Min: 1 Max: 15	8
Constraint/Dependency	This parameter is available only when QoS parameter is enabled.			
Customer Description	DMI QoS threshold value. Traffic with QoS equal to or above this value are considered as high priority hard real time traffic.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

960

TABLE 13-23: NDMIWTTQOSRSV PARAMETERS

Parameter Name	nDmiWttQoSRsv			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		Min: 1 Max: 64	Min: 1 Max: 32	1
Constraint/Dependency	This parameter is available only when QoS is enabled.			
Customer Description	WTT entries in DMI reserved for high priority hard real time traffic.			
Engineering Description	Maximum acceptable value must be minimum of WTT size - 1 or size of DMI non-coherent write data buffer or Coherent write data buffer. • Non-Coherent write data buffer is represented by DMI RB credits. • Coherent write data buffer is represented by number of connected DCEs multiplied by DCE RB credits per DMI. Max value = minimum of (Max WTT size - 1, DMI RB Credits - 1, DCE RB Credits - 1 * number of connected DCEs) Example: WTT size = 16 DMI RB credits = 24(non-Coherent write data buffer size) DCE RB credits = 4 and number of DCEs connected to DMI = 2. This gives the coherent write data buffer size of 4*2 = 8 As of the three numbers Coherent write data buffer size of 8 is smallest then maximum possible value is 8 - 1 = 7			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	User-GUI	
Change History				

Field Code Changed

963

TABLE 13-24: NDMIRTTQOSRSV PARAMETERS

Parameter Name	nDmiWttQoSRsv			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		Min: 1 Max: 64	Min: 1 Max: 32	1
Constraint/Dependency	This parameter is available only when QoS is enabled.			
Customer Description	RTT entries in DMI reserved for high priority hard real time traffic.			
Engineering Description	Maximum acceptable value must be RTT size - 1			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.4	User-GUI	
Change History				

Field Code Changed

966

13.7. DMI AddressBits parameter for custom AxiId address bit select

969 TABLE 13-25: ADDRESSBITS PARAMETER

Parameter Name	addressBits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer Array			
Constraint/Dependency				
Customer Description	This feature specifies an array of integers that consists of the bit indexes in an AXI transaction that can be used to encode the corresponding AXI ID. It applies only to writes.			
Engineering Description	Selected address bits will be used to generate corresponding axi ID. Minimum Array Size: 0 Maximum Array Size: wAwid Default Array Size: 0 (no entries) Array entry integer range: 0 to (system.concertocparams.wAddr-1) addressBits = addressIdMap.addressBits For Reads: 1. Fill the bottom bits with the corresponding address bit for example: a. Arid[0] = addressBits[0] b. Arid[1] = addressBits[1] … 2. If addressBits is an empty array this step is skipped. If there are more address bits defined than the ID width the bits above the ID width are ignored. 3. Starting from where above left off fill the rest of the bits with the bits above the cacheline. For example: a. Arid[x] = addressBits[wCachelineOffset] b. Arid[x+1] = addressBits[wCachelineOffset+1] … For Writes: 1. Fill the bottom bits with the corresponding address bit for example: a. Awid[0] = addressBits[0] b. Awid[1] = addressBits[1] … 2. If addressBits is an empty array this step is skipped. If there are more address bits defined than the ID width the bits above the ID width are ignored. 3. Starting from where above left off fill the rest of the bits with the bits above the cacheline plus two. For example: a. Awid[x] = addressBits[wCachelineOffset] b. Awid[x+1] = addressBits[wCachelineOffset] …			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.6.2	User-GUI	
Change History				

Field Code Changed

14. DII User Settable Parameters

14.1. DII resource parameters

TABLE 14-1: NRTTCTRLENTRY PARAMETERS

Parameter Name	nRttCtrlEnries			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		Min: 4 Max: 32	Min: 4 Max: 32	4
Constraint/Dependency				
Customer Description	Specify number of outstanding read transactions on the AXI interface.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2		
Change History				

Field Code Changed

TABLE 14-2: NWTTCTRLENTRY PARAMETERS

Parameter Name	nWttCtrlEnries			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		Min: 4 Max: 32	Min: 4 Max: 32	4
Constraint/Dependency				
Customer Description	Specify number of outstanding write transactions on the AXI interface.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2		
Change History				

Field Code Changed

TABLE 14-3: NLARGESTENDPOINT PARAMETER

Parameter Name	nLargestEndpoint			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		Min: 4 Max: 2^39	Min: 4 Max: 2^39	4
Constraint/Dependency				
Customer Description	Specify the size of the largest endpoint device connected to this DII. The size is in KBs. This size will be used to achieve endpoint ordering as defined by CHI architecture requirements.			
Engineering Description	Such a large value is used to connect with an FlexNoC and create a large endpoint space.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2		

Field Code Changed

Change History	

TABLE 14-4: NDIIRBCREDITS PARAMETER

Parameter Name	nDiiRbCredits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		Min: 2 Max: 32	Min: 2 Max: 32	2
Constraint/Dependency				
Customer Description	Specify the maximum number of non-coherent write request buffer credits.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2		
Change History				

Field Code Changed

TABLE 14-5: NCMDSKIDBUFSIZE PARAMETER

Parameter Name	**nCMDSkidBufSize**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 4 Max: ~~768~~8448	Min: 4 Max: ~~768~~8448	16
Constraint/Dependency	Restrict granularity, supporting only sizes: nCMDSkidBufArb + {0, 4, 8, 16, 32, 64, 96, 128, 160, 192, 224, 256, 512, 1024,2048, 4096, 8192}			
Customer Description	Total depth of skid buffer for commands to DCE/DII and the non-coherent port of DMI. The skid buffer is used to stage transaction requests from initiator agents. The number of required entries may be determined by traffic requirements and analysis using performance modeling. This value sets the total budget of protocol credits available for distribution.			
Engineering Description	This value sets the total budget of protocol credits available for distribution to communicating initiators. It is recommended to allow at least 2 credits for each active connection. CSR Address: 0xFF0 For a specific DII inside of a Ncore, the maximum value of nCMDSkidBufSize should be the total sum of nDiiCmdCredits as defined in Table 9 8 from any connected CAIU(s) and nDiiCmdCredits as defined in Table 14 7 from any connected NCAIU(s). During the configuration, It is recommended to do a sanity check to the value of nCMDSkidBufSize for any DII with a formula of (2*(the number of connected CAIUs + the number of connected NCAIUs)) , which assuming the minimum nDiiCmdCredits of 2 from each connected agent			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	Maximum value is increased from 512 to 768 in v3.7 Max increased to 8448 in 3.8			

Formatted: Font: (Default) Arial

Field Code Changed

TABLE 14-6: NCMDSKIDBUFARB PARAMETER

Parameter Name	**nCMDSkidBufArb**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 4 Max: 256	Min: 4 Max: 256	16
Constraint/Dependency	≤ nCMDSkidBufSize			

	restricted granularity, support only sizes: 4, 8, 16, 32, 48, 64, 128, 192, 256
Customer Description	Depth of skid buffer visible to arbitration. This value determines the size of the arbitration window within which arriving requests are selected based on QoS, priority and arrival time. It is recommended to start with a reasonably value for performance analysis - the area of a skid buffer grows with the square of this number and larger options will also significantly impact timing.
Engineering Description	This value sets the number of entries within the skid buffer that is visible to arbitration CSR Address: 0xFF0
Release Info	**Status** / **Effective version** / **Visibility** Active / 3.2 / User-GUI
Change History	Maximum value is support up to 256 from 64 in v3.6

Field Code Changed

990 TABLE 14-7: NEXCLUSIVEENTRIES PARAMETER

Parameter Name	**nExclusiveEntries**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 8	Min: 0 Max: 8	0
Constraint/Dependency				
Customer Description	defines the number of exclusive monitors			
Engineering Description	A value of 0 means no exclusive monitor will be instantiated			
Release Info	**Status** / **Effective version** / **Visibility** Active / 3.6 / User-GUI			
Change History				

Field Code Changed

14.2. DII address map parameters

993

TABLE 14-8: NADDRTRANSREGISTERS PARAMETER FOR DII

Parameter Name	**nAddrTransRegisters**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		Min: 0 Max: 16	Min: 0 Max: 8	4
Constraint/Dependency				
Customer Description	Specifies the number of address translation registers that are available within the DII. Refer to the address translation capability section in the reference manual.			
Engineering Description	[XXX] From NCore's spec, the limitation is 8. Need confirmation regarding range			
Release Info	**Status** / **Effective version** / **Visibility** Active / 3.2 /			
Change History				

Field Code Changed

14.3. DII performance monitor parameters

999 TABLE 14-9: DII NPERFCOUNTERS PARAMETERS

Parameter Name	nPerfCounters			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 16	Min: 0 Max: 16	0
Constraint/Dependency	Valid values supported are: 0, 4, 8 and 16			
Customer Description	Customer Description Total number of performance counter in a DII.			
Engineering Description	Architechture team would modify this as a common parameter in next NCore versions.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Release	
Change History				

Field Code Changed

Once the nPerfCounters is configured with a value bigger than zero, there will be 16 Latency Counters connected automatically by the hardware. And if the nPerfCounters is configured with a value of zero, NO Latency counters will be connected by the hardware, which implies the performance monitoring feature is completely disabled for this DII.

TABLE 14-10: DII LATENCY COUNTER PARAMETERS

Parameter Name	nLatencyCounters			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 32	Min: 0 Max: 16	16
Constraint/Dependency	Only two valid values are supported 0 or 16. A non-zero value is possible only if nPerfCounters is greater than or equal to 4.			
Customer Description	Number of Latency counters in a DII.			
Engineering Description	Parameter applies only to AIUs, DMIs and DIIs only and can be set individually.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Release	
Change History				

Field Code Changed

14.4. DII AddressBits parameter for custom AxiId address bit select

TABLE 14-11: ADDRESSBITS PARAMETER

Parameter Name	addressBits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer Array			
Constraint/Dependency				
Customer Description	This feature specifies an array of integers that consists of the bit indexes in an AXI transaction that can be used to encode the corresponding AXI ID.			
Engineering Description	Selected address bits will be used to generate corresponding axi ID. Minimum Array Size: 0 Maximum Array Size: min(wArId, wAwid) Default Array Size: 0 (no entries) Array entry integer range: 0 to (system.concertocparams.wAddr-1) addressBits = addressIdMap.addressBits For Reads: 1. Fill the bottom bits with the corresponding address bit for example: a. Arid[0] = addressBits[0] b. Arid[1] = addressBits[1] … 2. If addressBits is an empty array this step is skipped. If there are more address bits defined than the ID width the bits above the ID width are ignored. 3. Starting from where above left off fill the rest of the bits with the bits above the cacheline. For example: a. Arid[x] = addressBits[wCachelineOffset] b. Arid[x+1] = addressBits[wCachelineOffset+1] … For Writes: 1. Fill the bottom bits with the corresponding address bit for example: a. Awid[0] = addressBits[0] b. Awid[1] = addressBits[1] …			

	2. If addressBits is an empty array this step is skipped. If there are more address bits defined than the ID width the bits above the ID width are ignored. 3. Starting from where above left off fill the rest of the bits with the bits above the cacheline plus two. For example: a. Awid[x] = addressBits[wCachelineOffset+2] b. Awid[x+1] = addressBits[wCachelineOffset+1+2] …		
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.6.1	User-GUI
Change History			

Field Code Changed

1011

15. *DVE User Settable Parameters*

15.1. DVE resource parameters

Credit parameters and trace buffer parameters are defined at system level. Only SRAM selection will be configured in block level.

TABLE 15-1: MEMORY PARAMETER FOR DVE

Parameter Name	Memory			
Value	**Data Type**	**Architecture**	**Release**	**Default**
Constraint/Dependency				
Customer Description				
Engineering Description	This parameter is to assign SRAM. For the memory setting, refer **Error! Reference source not found.**			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 15-2: DVE EVENTBROADCASTERFIFODEPTH PARAMETER

Parameter Name	EventBroadcasterFIFOdepth			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	integer	Sum(useSysReqSender)	Sum(useSysReqSender)	Sum(useSysReqSender)
Constraint/Dependency	Add up total number of useSysReqSender of every units			
Customer Description				
Engineering Description	Used to size FIFO of Event Broadcaster hardware in the DVE			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

15.2. DVE performance monitor parameters

TABLE 15-3: DVE NPERFCOUNTERS PARAMETERS

Parameter Name	nPerfCounters			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 16	Min: 0 Max: 16	4
Constraint/Dependency	Valid values supported are: 0,4,8 and 16			
Customer Description	Total number of performance counter in Ncore Unit			
Engineering Description	Archi team would modify this as a common parameter in next NCore versions.			

Release Info	Status	Effective version	Visibility
	Active	3.2	User-GUI
Change History			

Field Code Changed

Please note that latency counters are **NOT** used for a DVE. Therefore, there isn't any latency counter is connected no matter the number of performance counter is set as zero, or bigger. This means that the latency counters are only configurable in AIUs, DMIs and DIIs.

15.3. DVM System level parameters

Ncore provides two system-level configurable parameters for DVM operation: nDvmCmdCredits and nDvmSnpCredits. nDvmCmdCredits allows the AIU to have that many non-Sync/Sync DVMOps outstanding to DVE.

nDvmSnpCredits allows the DVE to issue that many DVMOps for snoops outstanding at a time. This value is the minimum of such outstanding DVMInv snoops supported by any AIU in the system. Note that each DVMOp snooped corresponds to 2 SNPreq messages.

The parameter noDVM value defined in 3.10 , which globally turns off the DVM functionalities throughout an Ncore

TABLE 15-4: DVMVERSIONSUPPORT PARAMETER

Parameter Name	DVMVersionSupport			
Value	Data Type	**Architecture**	**Release**	**Default**
	Integer	{8,0}, {8,1}, {8,4}	{8,0}, {8,1}, {8,4}	{8,4}
Constraint/Dependency	Refer to table 20, and based on the interface find the maximum common capability of all AIU interface.			
Customer Description	DVM version capability of the system. The value is suggested for the User to configure the system.			
Engineering Description	Pass the parameter to register DVEUDVMRR "DVM Revision Register" in DVE register space The version number is encoded as the concatenation of two 4 bit integers {4'd,4'd}. The first integer represents the main DVM version and the second the subversion number. For example: DVM_v8.1 the version number is {4'd8,4'd1} or {8,1}			
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 15-5: NDVMCMDCREDITS PARAMETER

Parameter Name	nDvmCmdCredits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	*Min: 2* *Max: 4*	*Min: 2* *Max: 4*	2
Constraint/Dependency	Must be a multiple of 2.			
Customer Description	Number of DVM command credits between an AIU and a DVE.			
Engineering Description	This parameter is applicable to all AIUs that can issue DVMs.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 15-6: NDVMSNPCREDITS PARAMETER

Parameter Name	nDvmSnpCredits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Enum	Equal to the result of { noDVM? 0: Max {8, (total number of DVM agents + 1) * 2}}	Max {8, (total number of DVM agents + 1) * 2}	0 or 8 depends on whether noDVM is set or not
Constraint/Dependency	Must be a multiple of 2			
Customer Description	If noDVM[5] system wide parameter is not set, take the maximum value out of 8 or (total number of DVM agents + 1) * 2			
Engineering Description	If noDVM system wide parameter is not set, take the maximum value out of 8 or (total number of DVM agents + 1) * 2			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

[5] It is the noDVM value defined in 3.10 , which globally turns off the DVM functionalities throughout an Ncore

16. *NCAIU User Settable Parameters*

16.1. NCAIU multiport parameters

TABLE 16-1: NNATIVEINTERFACEPORTS PARAMETER FOR NCAIU

Parameter Name	nNativeInterfacePorts			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 8	Min: 1 Max: 8	1
Constraint/Dependency	Valid values are power of 2s. 1, 2, 4, or 8.			
Customer Description	Specifies the number of native interface ports			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 16-2: ANCAIUINTVFUNC PARAMETER FOR NCAIU

Parameter Name	aNcaiuIntvFunc			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Array of integers	*aPrimaryBits*, aSecondaryBits	*aPrimaryBits*, Not user visible	
Constraint/Dependency	aPrimaryBits depth depends on nNativeInterfacePorts parameter value, it will be log2 of nNativeInterfacePorts. The bits must be address bits between Max address width minus 1 and cacheline boundary address bit. For 64Bcache line it is 6. Example aPrimaryBits : [9, 8, 6] aSecondaryBits is an array of string, its depth will be same as aPrimaryBits. The string represents a hexadecimal number one hot encoded. Bits selected here cannot be same as the bits in aPrimaryBits. Example aSecondaryBits: [“'h4000”, “'h0”, “'h800”]			
Customer Description	aPrimaryBits depth depends on nNativeInterfacePorts parameter value, it will be log2 of nNativeInterfacePorts. The bits must be address bits between Max address width minus 1 and cacheline boundary address bit. For 64Bcache line it is 6. Example aPrimaryBits : [7, 6] for a 4 ports NCAIU interleaving at 64B address boundary.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

16.2. NCAIU resource parameters

TABLE 16-3: NOTTCTRLENTRIES FOR NCAIU

Parameter Name	nOttCtrlEntries			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 8 Max: 1024	Min: 8 Max: 1024	32
Constraint/Dependency	Must be multiple of nNativeInterfacePorts. When divided by nNativeInterfacePorts it must be less than or equal to 128. Min is for a single port. Max is for nNativeInterfacePorts=8			
Customer Description	Specify the maximum number of outstanding native transactions this AIU should support.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 16-4: MEMORY PARAMETER FOR NCAIU

Parameter Name	Memory			
Value	**Data Type**	**Architecture**	**Release**	**Default**
Constraint/Dependency				
Customer Description				
Engineering Description	For the memory setting, refer to Chapter 24.2 for detail			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

16.3. NCAIU credit parameters

TABLE 16-5: NDCECMDCREDITS FOR NCAIU

Parameter Name	nDceCmdCredits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 2 Max: 32	Min: 2 Max: 32	32
Constraint/Dependency	Must be multiple of nNativeInterfacePorts for both min and max ranges and actual value. Min is for a single port.			
Customer Description	Specify the maximum number of credits for coherent transactions per DCE. This should be determined based on required bandwidth and network round trip latency.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-Register	

Field Code Changed

Change History	Credit is software programmable since 3.4 (Visibility: User-GUI → User-Register)

TABLE 16-6: NDMICMDCREDITS FOR NCAIU

Parameter Name	nDmiCmdCredits			
Value	Data Type	Architecture	Release	Default
	Integer	Min: 2 Max: 32	Min: 2 Max: 32	16
Constraint/Dependency	Must be multiple of nNativeInterfacePorts for both min and max ranges and actual value. Min is for a single port.			
Customer Description	Specify the maximum number of credits for non-coherent transactions per DMI. This should be determined based on required bandwidth and network round trip latency.			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-Register	
Change History	Credit is software programmable since 3.4 (Visibility: User-GUI → User-Register)			

Field Code Changed

TABLE 16-7: NDIICMDCREDITS FOR NCAIU

Parameter Name	nDiiCmdCredits			
Value	Data Type	Architecture	Release	Default
	Integer	Min: 2 Max: 32	Min: 2 Max: 32	16
Constraint/Dependency	Must be multiple of nNativeInterfacePorts for both min and max ranges and actual value. Min is for a single port.			
Customer Description	Specify the maximum number of credits for non-coherent transactions per DII. This should be determined based on required bandwidth and network round trip latency.			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History	Credit is software programmable since 3.4 (Visibility: User-GUI → User-Register)			

Field Code Changed

16.4. NCAIU address map parameter

TABLE 16-8: FNCSRACCESS_PARAMETER

Parameter Name	fnCsrAccess			
Value	Data Type	Architecture	Release	Default
	Valid Values	True, False	True, False	false
Constraint/Dependency	Should be true on at least one AIU. Always false on coherent AXI NCAIU where nonCoherentMode parameter is set to false.			
Customer Description	Enables CSR access via this AIU			

Engineering Description			
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	User-GUI
Change History			

Field Code Changed

16.5. NCAIU snoop filter parameters

TABLE 16-9: SNOOPFILTERASSIGNMENT PARAMETER

Parameter Name	**snoopFilterAssignment**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 64	Min: 0 Max: 64	0
Constraint/Dependency	Only visible for (1) "AXI" with "hasProxyCache == TRUE". Also refer table in chapter 9.4.			
Customer Description	Specify the snoop filter associated with this AIU. This only applies for AIUs with proxy cache and ACE interface.			
Engineering Description	Will stay at AIU parameter at least for NCore 3.2. Already had agreed, and too late to change.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

16.6. NCAIU proxy cache parameters

TABLE 16-10: PROXY CACHE ENABLE PARAMETERS

Parameter Name	hasProxyCache			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	True, False	True, False	False
Constraint/Dependency	This option enables a ProxyCache is configured for a NCAIU.			
Customer Description	This option adds a ProxyCache in NCAIU.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 16-11: PROXY CACHE NTAGBANK CONFIGURATION PARAMETERS

Parameter Name	**nTagBanks**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 2	Min: 1 Max: 2	1
Constraint/Dependency	Values limited to 1, 2			
Customer Description	Number of Tag banks.			

Engineering Description			
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	User-GUI
Change History			

Field Code Changed

TABLE 16-12: PROXY CACHE NDATABANK CONFIGURATION PARAMETERS

Parameter Name	**nDataBanks**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 4	Min: 1 Max: 4	1
Constraint/Dependency	Values limited to 1, 2.4			
Customer Description	Number of Data banks.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 16-13: PROXY CACHE NSETS CONFIGURATION PARAMETERS

Parameter Name	**nSets**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 16 Max: 8192	Min: 2 Max: 8192	16
Constraint/Dependency	Number of sets must be power of 2 number			
Customer Description	Number of sets for proxy cache			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 16-14: PROXY CACHE NWAYS CONFIGURATION PARAMETERS

Parameter Name	**nWays**			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 2 Max: 16	Min: 2 Max: 16	2
Constraint/Dependency	Available number of ways for a proxy cache are: 2, 4, 8, and 16			
Customer Description	Number of ways for proxy cache			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

16.7. NCAIU performance counter parameters

TABLE 16-15: NPERFCOUNTERS PARMAETER FOR NCAIU

Parameter Name	nPerfCounters			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 16	Min: 0 Max: 8	0
Constraint/Dependency	Only three valid values are supported. 0, 4 and 8			
Customer Description	Total number of performance counter in a NCAIU.			
Engineering Description	Archi team would modify this as a common parameter in next NCore versions.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

Once the nPerfCounters is configured with a value bigger than zero, there will be 16 Latency Counters connected automatically by the hardware. And if the nPerfCounters is configured with a value of zero, NO Latency counters will be connected by the hardware, which implies the performance monitoring feature is completely disabled for this IOAIU. nPerfCounters value of 16 triggers an implementation issue.

TABLE 16-16: NLATENCYCOUNTERS PARMAETER FOR NCAIU

Parameter Name	nLatencyCounters			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 32	Min: 0 Max: 16	16
Constraint/Dependency	Only two valid values are supported 0 or 16. A non-zero value is possible only if nPerfCounters is greater than or equal to 4.			
Customer Description	Number of Latency counters in a NCAIU.			
Engineering Description	Parameter applies only to AIUs, DMIs and DIIs only and can be set individually.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

16.8. NCAIU disable read data interleaving parameters

TABLE 16-17: FNDISABLERDINTERLEAVE PARMAETER FOR NCAIU

Parameter Name	fnDisableRdInterleave			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 1	Min: 0 Max: 1	0

Constraint/Dependency			
Customer Description	When set disables read data interleaving across different AXI IDs		
Engineering Description	When set disables read data interleaving across different AXI IDs. This parameter applies to NCAIU with AXI, ACE-LITE and ACE5-LITE ports		
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.4	User-GUI
Change History			

Field Code Changed

16.9. NCAIU SysCmd Hardware parameters

TABLE 16-18: USESYSCOENGINE PARAMETERS FOR NCAIU

Parameter Name	useSysCoEngine			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency	Always True for ACE/CHI/AXI with Proxy Cache AIUs if useSysCoInt is True, set True to this parameter			
Customer Description				
Engineering Description	Used to instantiate SysCo Engine hardware in the AIU			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 16-19: USESYSREQSENDER PARAMETERS FOR NCAIU

Parameter Name	useSysReqSender			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency	Always True for ACE/CHI AIUs Optional for ACE_Lite + DVM AIUs if useEventOutInt is True, set True to this parameter			
Customer Description				
Engineering Description	Used to instantiate SysReq Sender hardware in the AIU			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 16-20: USESYSREQRECEIVER PARAMETERS FOR NCAIU

Parameter Name	useSysReqReceiver			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	*True, False*	*True, False*	True
Constraint/Dependency	Always True for ACE/CHI AIUs if useEventInInt is True, set True to this parameter			
Customer Description				
Engineering Description	Used to instantiate SysReq Receiver hardware in the AIU			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

16.10. NCAIU Connectivity parameters

TABLE 16-21: HEXAIUDCEVEC PARAMETERS FOR NCAIU

Parameter Name	hexAiuDceVec			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: FFFFFFFF	Min: 0 Max: FFFFFFF	1
Constraint/Dependency	Size of the vector is equal to the number of DCEs in the system. Every bit in the vector that is set to one represents a DCE at that NodeID that is connected to the AIU.			
Customer Description				
Engineering Description	This must be a port in RTL (tACHL) and tie off parameter in SW Every bit in the vector that is set to one specifies that the particular AIU is connected to the associated DCE at that NunitID			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 16-22: HEXAIUDMIVEC PARAMETERS FOR NCAIU

Parameter Name	hexAiuDmiVec			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: FFFFFFFF	Min: 0 Max: FFFF	1
Constraint/Dependency	Size of the vector is equal to the number of DMIs in the system. Every bit in the vector that is set to one represents a DMI at that NodeID that is connected to the AIU.			
Customer Description				
Engineering Description	This must be a port in RTL (tACHL) and tie off parameter in SW Every bit in the vector that is set to one specifies that the particular AIU is connected to the associated DMI at that NunitID			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 16-23: HEXAIUDIIVEC PARAMETERS FOR NCAIU

Parameter Name	hexAiuDiiVec			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: FFFFFFFF	Min: 0 Max: FFFF	1
Constraint/Dependency	Size of the vector is equal to the number of DIIs in the system. Every bit in the vector that is set to one represents a DII at that NodeID that is connected to the AIU.			
Customer Description				
Engineering Description	This must be a port in RTL (tACHL) and tie off parameter in SW Every bit in the vector that is set to one specifies that the particular AIU is connected to the associated DII at that NunitID			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	

Field Code Changed

Change History	

1131

TABLE 16-24: HEXAIUCONNECTEDDCEFUNITID PARAMETERS FOR NCAIU

Parameter Name	hexAiuConnectedDceFunitId			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: FFFFFFFF	Min: 0 Max: FFFF	1
Constraint/Dependency	List of DCE Funit IDs that are connected to the AIU. This list can be ordered in Nunit ID order.			
Customer Description				
Engineering Description	This must be a port in RTL (tACHL) and tie off parameter in SW. List of DCE FuntilDs that are connected to the AIU.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 16-25: NAIUCONNECTEDDCES PARAMETERS FOR NCAIU

Parameter Name	nAiuConnectedDces			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 64	Min: 1 Max: 32	1
Constraint/Dependency	Number of DCEs connected to this each AIU.			
Customer Description				
Engineering Description	Specifies the number of caching agents (AIUs) that are connected to DCE			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

16.11. NCAIU OWO parameters

TABLE 16-26: ORDEREDWRITEOBSERVATION PARAMETER

Parameter Name	orderedWriteObservation			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	False
Constraint/Dependency				
Customer Description	The Ordered Write Observation feature to accelerate PCIe traffic			
Engineering Description	This parameter should be set as true once Ordered Write Observation feature is required, which mostly works directly for PCIe traffic acceleration. Once this feature is enabled, the interface is only allowed to be either ACE-Lite or AXI and proxy cache can Not be configured together with this option. Please refer to section 4.14 of the system architecture specification for more limitations and constraints.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7.1	User-GUI	
Change History				

Field Code Changed

TABLE 16-27: MULTICYCLEODSRAM PARAMETER

Parameter Name	multicycleODSRAM			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True/False	True/False	False
Constraint/Dependency				
Customer Description	OD timing closure optimization parameter for OWO configuration			
Engineering Description	This parameter only takes effect and be visible if orderedWriteObservation parameter is set to TRUE. It is mainly used to ease timing closure for an IOAIUp once OWO is enabled and the Outstanding Data (OD) Buffer is big.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7.1	User-GUI	
Change History				

Field Code Changed

1143 TABLE 16-28: IOAIUPDATAWIDTH PARAMETER

Parameter Name	IOAIUpDataWidth			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Enum	{256, 512}	{256, 512}	256
Constraint/Dependency	This parameter only takes effect and be visible if orderedWriteObservation parameter is set to TRUE. An IOAIU configuration with OWO enabled has a default ingress data width of 256 bit. When it is set to 512-bit, it implies the ingress interface to the IOAIUp is running at 1GHz. Thus, Maestro needs to insert an asynchronous gasket right in front a data width adaption gasket to interface the PCIe traffic properly from the 3rd party PCIe controller. If this parameter is set to 256 bit, it implies a synchronous interface from the 3rd party PCIe controller to the IOAIUp and both the asynchronous gasket and the data width adaption gasket are not needed.			
Customer Description	Define the data width of the master interface from the 3rd PCIe controller			
Engineering Description	Define the data width of the master interface from the 3rd PCIe controller			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.7.1	User-GUI	
Change History				

Field Code Changed

Once IOAIUpDataWidth parameter is set to 512, whatever value taken in Ncore system level syncDepth parameter defined in
1146 Table 3-27: syncDepth parameter~~Table 3-27: syncDepth parameter~~ also takes effect to all the synchronizers employed inside the asynchronous frequency adaption gasket inserted automatically by Maestro.

1149 TABLE 16-29: NENTRIESINSHIM PARAMETER

Parameter Name	nEntriesInShim			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 4 Max: 1024	Min: 4 Max: 1024	32
Constraint/Dependency	This parameter only takes effect and be visible if orderedWriteObservation parameter is set to TRUE **AND** IOAIUpDataWidth is set to 512..			
Customer Description	The number of entries the axi_shim instantiates inside of the data width adaption gasket. It should be equal to the maximum outstanding transactions an IOAIUp can support.			
Engineering Description	This parameter defines the number of entries of the axi_shim that instantiates inside of the data width adaption gasket. It should be equal to the maximum outstanding transactions an IOAIUp can support, which should be the same value as what it is defined for the nOttCtrlEntries. And it will be derived internally once a 512 bit interface is used			

Release Info	Status	Effective version	Visibility
	Active	3.7.1	Engineering
Change History			

Field Code Changed

17. *NCAIU non-User settable parameters*

TABLE 17-1: NSTTCTRLENTRIES FOR NCAIU

Parameter Name	nSttCtrlEntries			
Value	Data Type	Architecture	Release	Default
	Integer	Derived	Derived	Derived
Constraint/Dependency	If Proxy-cache or OrderedWriteObservation : $\sum_{nDCEs} nAiuSnpCredits + (nDies - 1) \times nRemoteSnpCredits$ If ACE5-Lite and DVM : $nDvmSnpCredits$ Else 0			
Customer Description	Specifies the maximum number of outstanding snoop transactions the AIU can receive.			
Engineering Description	Specifies the maximum number of outstanding snoop transactions the AIU can receive.			
Release Info	Status	Effective version	Visibility	
	Preview	3.8	Engineering	
Change History				

Field Code Changed

18. *GIU user Settable parameters*

18.1. Virtual channel credits.

TABLE 18-1 : VC_DESCRIPTOR.NUMBEROFCREDIT PARAMETER

Parameter Name	VC_descriptor NumberOfCredit			
Value	Data Type	Architecture	Release	Default
	Array[integer]	Min: [4,4,4,4] Max: [31,31,31,31]	Min: [4,4,4,4] Max: [31,31,31,31]	[15,15,15,15]
Constraint/Dependency	Length of the array is 4 and for each entry min : 4, max : 31.			

Customer Description	Each network maps to its own virtual channel. This is the number of protocol credit for each virtual channel.		
Engineering Description	Each network maps to its own virtual channel. This is the number of protocol credit for each virtual channel. It is recommended that the user does not provide an array but individual number for each entry of the array. Proposed structure: Number of credit for CN0, CN1, CN3 and DN is mapped to array [a, b, c, d] respectively.		
Release Info	**Status**	**Effective version**	**Visibility**
	Preview	3.8	Engineering
Change History			

Field Code Changed

19. *GIU non-user settable parameter*

The following parameters are parameters used in the GIU and are not user settable/visible.

TABLE 19-1: VC_DESCRIPTOR PARAMETER

Parameter Name	vc_descriptor			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Object			
Constraint/Dependency				
Customer Description				
Engineering Description	This object contains the parameters related to the VC used on to put concerto messages on a single CXS link. { "nVC" : 4, "wCredit" : 4, "NumberOfCredit" : <Table 18-1~~Table 18-1~~>, "MessageSizeInGranule" : [3,2,2,10] }			
Release Info	**Status**	**Effective version**	**Visibility**	
	Preview	3.8	Engineering	
Change History				

Field Code Changed

TABLE 19-2 :PACKET_DESCRIPTOR PARAMETER

Parameter Name	packet_descriptor			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Object			
Constraint/Dependency				
Customer Description				
Engineering Description	This object contains the parameters related to the packet format used to put concerto messages on a CXS link. { "SMI_ndp_packing_order" : ["msg_pri","ndp_len","msg_type","targ_id","src_id","msg_id","ndp","msg_user","dp_present"],			

	"SMI_dp_packing_order" : ["data","user"], "NumberOfGranulePerBeat" : 6, "GranuleSizeInBytes" : 10, "NumberOfBeatPerContainer" :4, "StartBits" : [{"GranuleId" : [511,510,509,508,507,506]},{"GranuleId" : [15,14,13,12,11,10]}, {"GranuleId" : [511,510,509,508,507,506]},{"GranuleId":[15,14,13,12,11,10]]}, "PayloadBits" : [16,16,16,16], "CreditreturnBits" : [{"values" : [0,496,3]},{"values" :[0,501,3]},{"values" : [1,0,3],{"values" : [1,5,3]}] }

Release Info	Status	Effective version	Visibility
	Preview	3.8	Engineering
Change History			

Field Code Changed

1173

20. *Cache and snoop filter User Settable Parameters*

The following parameters apply to caches and snoop filters. The CCP is a configurable Cache IP block. It is commonly used for all the IPs which requires Cache access. Currently it is being used by Proxy Cache in IO-AIU and SMC in DMI. The snoop filter is in DCE.

TABLE 20-1: NSETS PARAMETERS OF CCP

Parameter Name	nSets			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	16', '32', '64', '128', '256', '512', '1024', '2048', '4096', '8192	16', '32', '64', '128', '256', '512', '1024', '2048', '4096', '8192	16
Constraint/Dependency	The number of sets per data bank must be greater than the number of data banks. The number of sets per tag bank must be greater than the number of tag banks.			
Customer Description	Specify the number of sets/entries in the Cache.			
Engineering Description	Expect log2(nSets) bits for primary selection bits. Must be multiple of nNativeInterfacePorts for both min and max ranges and actual value.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 20-2: NWAYS PARAMETER OF CCP

Parameter Name	nWays			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		Min:2 Max: 16	Min:2 Max: 16	2
Constraint/Dependency				
Customer Description	Specify the number of sets/entries in the Cache.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2		
Change History				

Field Code Changed

TABLE 20-3: USESCRATCHPAD PARAMETER OF CCP

Parameter Name	useScratchPad			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	FALSE
Constraint/Dependency				
Customer Description	Enable Scratchpad. The visibility will be overridden based on block type.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

Table 20-4: PriSubDiagAddrBits parameters

Parameter Name	PriSubDiagAddrBits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Array of strings			
Constraint/Dependency				
Customer Description	Specify address bits to be used as primary set select bits.			
Engineering Description	Prisubdiagaddrbits depth must be log2 (nSets/nNativeInterfacePorts). The bits must be address bits between Max address width minus 1 and cacheline boundary address bit. For 64Bcache line it is 6.They cannot include address bits used in aPrimaryBits of aNcaiuIntvFunc and address bits used in aPrimaryAiuPortBits			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

1188

TABLE 20-5: TAGBANKSELBITS PARAMETERS

Parameter Name	TagBankSelBits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Array of strings			
Constraint/Dependency				
Customer Description	Tag bank selection bit			
Engineering Description	The tag bank selection bit values must be unique. The tag bank selection bit must be one of the primary set selection bits. The number of tag bank bits must be log2(nTagBanks).			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

1191 TABLE 20-6: DATABANKSELBITS PARAMETERS

Parameter Name	DataBankSelBits			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Array of strings			
Constraint/Dependency				
Customer Description	Specify data bank select bit. This bit must be one of the bits from the primary select bits.			
Engineering Description	The data bank selection bit values must be unique. The data bank bits must be one of the primary set selection bits. The number of data bank bits must be log2(nDataBanks) bits.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 20-7: CACHEREPLPOLICY PARAMETER

Parameter Name	cacheReplPolicy

Value	Data Type	Architecture	Release	Default
	Enum	RANDOM, NRU, SRRIP, pLRU	RANDOM, NRU, pLRU	RANDOM
Constraint/Dependency	Available in DMI with SMC enabled Available in IOAIU with ProxyCache enabled Available in DCE with Snoop Filters (only Random and pLRU are available)			
Customer Description	Cache Replacement Policy			
Engineering Description	Depending on the selected policy, a dependent parameter cacheReplStateWidth needs to be calculated . That parameter defines the number of bits required to represent the current position in the replacement algorithm for each cacheline in the set			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History	pLRU is supported since v3.6			

Field Code Changed

TABLE 20-8: CACHEREPLSTATEWIDTH PARAMETER

Parameter Name	cacheReplStateWidth			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	0, 1, 2	0, 1	0
Constraint/Dependency	Available in DMI with SMC enabled Available in IOAIU with ProxyCache enabled Available in DCE with Snoop Filters			
Customer Description				
Engineering Description	This parameter value is derived based on the cacheReplPolicy parameter: RANDOM: 0, NRU: 1, SRRIP: 2, pLRU: 1 Note: For the SSRIP implementation we may want to consider an optimization - reserving state 00 as indication of an invalid cache line can save one of the standard state bits that indicate valid, dirty.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

21. *Tiling Related Parameters*

TABLE 21-1: PORT_ID PARAMETER FOR PACKETIZER

Parameter Name	Port_ID			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Binary	{1'b0, 1'b1}	{1'b0, 1'b1}	1'b0
Constraint/Dependency	Port_ID is used together with FunitIDs for a packetizer when tiling is supported. It is a one-bit wide binary value and it is tied to 1'b0.			
Customer Description	When tiling is enabled for a packetizer, Maestro needs to concatenate FUnitID and Port_ID to make a tie-off value for the packetizer.			
Engineering Description	When tiling is enabled for a packetizer, Maestro needs to concatenate FUnitID and Port_ID to make a tie-off value for the packetizer.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.6	Engineering	
Change History				

Field Code Changed

TABLE 21-2: ROUTEONSID PARAMETER FOR PACKETIZER

Parameter Name	routeOnSid			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Boolean	True, False	True, False	False
Constraint/Dependency	Once the tiling is supported for a packetizer, routeOnSid needs to be set true by Maestro so the packetizer can work properly for a specific tile.			
Customer Description				
Engineering Description	The packetizer needs to support the option of concatenating SMI Source and Target IDs to be used as an input key to the LUT once the tiling is supported. This means the packetizer can use {SMI NDP SID, SMI NDP TID } as the input to the path LUT. A new boolean JS parameter, **routeOnSid**, needs to be set to true by Maestro software to enable this feature. SID stands for Source ID and TID stands for Target ID.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.6	Engineering	
Change History				

Field Code Changed

22. *Legato User Settable Parameters*

Async adapter and dw_adapter are automatically inserted. Async adapters are inserted between different clock domains, and dw_adpaters are inserted if there is mismatch between input and output of the link.

Data width inside of the network would be configured using portDataWidth of the sym_switch and sym_buf_switch. Network parameter is not being supported at NCore 3.6.

Some of the derived/fixed parameters have been described in this section (because many of the engineers are reading only user settable part) but they may be moved to a separate "derived/fixed" chapter in a later version of the specification.

22.1. sym_switch/sym_buf_switch

The sym_buf_switch supports configurable buffers at the ingress port of the switches.

TABLE 22-1: SYM_BUF_SWITCH AND SYM_SWITCH PARAMETER: PORTDATAWIDTH

Parameter Name	portDataWidth			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid values	64, 128, 256, 512	64, 128, 256, 512	256
Constraint/Dependency	Can only be 512 for the first switch connected to the GIU.			
Customer Description	Data width of all ports of the switch			
Engineering Description	Applied to sym_switch and sym_buf_switch			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 22-2: SYM_BUF_SWITCH PARAMETER: INPUTBUFFERDEPTH

Parameter Name	inputBufferDepth			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	[0, 2, 4, 8, 12, 16, 24, 32]	[0, 2, 4, 8, 12, 16, 24, 32]	2
Constraint/Dependency				
Customer Description	Buffer depth is buffer depth at **input port** (Layer 0) If we configure inputBufferDepth 0, **sym_switch** is configured.			
Engineering Description	NCore 3.6 supports only Buffer Layer 0 buffers.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

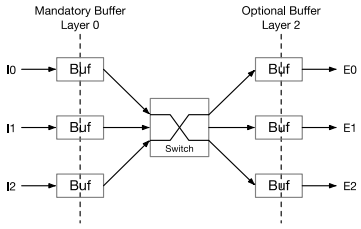


FIGURE 22-1: SYM_BUF_SWITCH IN CDTI, WITH ONLY ONE VC

22.2. sym_async_adapter

Clock adapters require the specification of two different FIFO depths:

- The depth of the synchronizers used for signals that cross domains for metastability reasons.
- The depth of the circular FIFO used to transfer data from one side to the other and the depth affects functional throughput.

The synchronizer depth is configurable to supporting a circular FIFO (added from Ncore 3.2)

- A new system parameter called **syncDepth** is added to configure synchronizer depth of sym_async_adapter. This new parameter will be used to set the depth of the synchronizers.
- Circular FIFO depth = Math.ceil(2*(syncDepth+1.5)).
- Ncore 3.6 supports syncDepth values of 2, 3, and 4 only

22.3. chi_async_adapter

sym_async_adapter is for SMI interface, and chi_async_adapter is to support CHI interface. It has a slave CHI interface and a master CHI interface, each interface has its own clock. Depth of the circular FIFO are calculated according to the number credit if the CHI interface. No user settable parameters.

22.4. sym_nRate_adapter

Please refer to section 26.5 for more details.

22.5. cxs_async_adapter

sym_async_adapter is for SMI interface, and cxs_async_adapter is to support cxsinterface. It has a slave CXS interface and a master CXS interface, each interface has its own clock. Depth of the circular FIFO are calculated according to the number credit if the CXS interface. No user settable parameters.

22.6. dw_adapter

No user settable parameter. Buffer depth is calculated inside of the block

If **pipeforward** and **pipeBackward** are set true, the depth parameters **dfDepth** and **hfDepth** must be set to at least 2, otherwise bubbles will be inserted into the data stream.

22.7. sym_pipe_adapter

TABLE 22-3: SYM_PIPE_ADAPTER PARAMETER: DEPTH

Parameter Name	depth			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	[0,1,2,3]	[1,2]	2
Constraint/Dependency				
Customer Description	Fifo depth inside of the sym_pipe_adapter			
Engineering Description	Depth 1 is expected for CSR network – mostly the network which doesn't require performance.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

23. *Derived/Fixed Socket Parameters*

23.1. AXI Interface

TABLE 23-1: AXI INTERFACE FIXED PARAMETERS

Parameter Name	Type		Default	Min	Max	Enum	Description
wResp	Integer	Fixed	2	N/A	N/A		
wWUser	*Integer*	Fixed	0	N/A	N/A	Not being used	
wBUser	Integer	Fixed	0	N/A	N/A	Not being used	
wRUser	Integer	Fixed	0	N/A	N/A	Not being used	
wLen	Integer	Fixed	8	N/A	N/A		
wSize	Integer	Fixed	3	N/A	N/A	Only 3. Because we are not supporting 512.	
wLock	Integer	Fixed	1	N/A	N/A	Always 1	
wQos	Integer	Fixed	0	N/A	N/A	['0', '4']	
wRegion	Integer	Fixed	0	N/A	N/A	Fixed as 0	
wProt	Integer	Fixed	3	N/A	N/A	Fixed as 3	

23.2. APB Interface

TABLE 23-2: APB INTERFACE FIXED PARAMETERS

Parameter Name	Type		Default	Min	Max	Enum	Description
wAddr	Integer	Fixed	12	N/A	N/A		
wData	*Integer*	Fixed	32	N/A	N/A		
wProt	Integer	Derived	0	0	3	['0', '3']	3 if APB4
wStrb	Integer	Derived	0	0	4	['0', '1', '2', '4']	4 if APB4
wPSlverr	Integer	Fixed	0	N/A	N/A		

23.3. ACE Interface

TABLE 23-3: ACE INTERFACE USER SETTABLE PARAMETERS

Parameter Name	Type		Default	Min	Max	Description/Derivation
eUnique	Integer	Eng. Param.	1	0	1	
wCdData	Integer	Fixed	0	N/A	N/A	
wSnoop	Integer	Eng. Param.	3	3	3	
eAc	Integer	Fixed	1	N/A	N/A	
wResp	Integer	Fixed	4	N/A	N/A	
eDomain	Integer	Fixed	1	N/A	N/A	
useQos	Boolean	Derived				useQoS: system parameter
wQos	Integer	Derived				wQos = (useQos) ? 4 : 0;

23.4. ACE5-LITE Interface

TABLE 23-4: ACE5-LITE INTERFACE DERIVED PARAMETERS

Parameter Name	Type		Default	Min	Max	Description/Derivation
wLoop	Integer	Fixed	0	N/A	N/A	
eTrace	Integer	Fixed	1	1	1	MAES-3605, changed from 0 to 1 to support Trace signal at NCore 3.6.
eUnique	Integer	Fixed	0	N/A	N/A	
wCdData	Integer	Fixed	0	N/A	N/A	
wSnoop	Integer	Derived	3	3	4	wSnoop = eStash == 1 ? 4 : 3;
eStash	Integer	Fixed	1	N/A	N/A	
eAtomic	Integer	Fixed	1	N/A	N/A	
eDomain	Integer	Fixed	1	N/A	N/A	

23.5. ACE-LITE Interface

1284 TABLE 23-5: ACE-LITE INTERFACE DERIVED PARAMETERS

Parameter Name	Type		Default	Min	Max	Description/Derivation
wLoop	wLoop	Fixed	0	N/A	N/A	
eTrace	eTrace	Fixed	0	N/A	N/A	
eUnique	eUnique	Fixed	0	N/A	N/A	
wCdData	wCdData	Fixed	0	N/A	N/A	
wSnoop	wSnoop	Fixed	3	N/A	N/A	
eStash	eStash	Fixed	0	N/A	N/A	
eAtomic	eAtomic	Fixed	0	N/A	N/A	
eDomain	eDomain	Fixed	1	N/A	N/A	

23.6. CHI_B Interface

1287

TABLE 23-6: CHI_B INTERFACE DERIVED PARAMETERS-1

Parameter Name	Type		Default	Min/ Max	Description/Derivation
SrcID	Integer	Derived	7	7/11	SrcID = NodeID_Width;
TgtID	Integer	Derived	7	7/11	TgtID = NodeID_Width;
TxnID	Integer	Fixed	8	N/A	
ReturnNID	Integer	Derived	7	7/11	ReturnNID = NodeID_Width;
StashNIDValid	Integer	Fixed	1	N/A	
ReturnTxnID	Integer	Fixed	8	N/A	
REQ_Opcode	Integer	Fixed	6	N/A	
RSP_Opcode	Integer	Fixed	4	N/A	
SNP_Opcode	Integer	Fixed	5	N/A	
DAT_Opcode	Integer	Fixed	3	N/A	
Size	Integer	Fixed	3	N/A	
wAddr	Integer	Derived	48	44/52	Physical address width from user input wAddr = {44, 48, 52}
NS	Integer	Fixed	1	N/A	
LikelyShared	Integer	Fixed	1	N/A	
AllowRetry	Integer	Fixed	1	N/A	
Order	Integer	Fixed	2	N/A	
PCrdType	Integer	Fixed	4	N/A	
MemAttr	Integer	Fixed	4	N/A	
SnpAttr	Integer	Fixed	1	N/A	
LPID	Integer	Fixed	5	N/A	
Excl	Integer	Fixed	1	N/A	
ExCompAck	Integer	Fixed	1	N/A	
TraceTag	Integer	Fixed	1	N/A	

Parameter Name	Type		Default	Min/ Max	Description/Derivation
DAT_RSVDC	Integer	Fixed	0	N/A	Not supported and is always fixed at zero
RespErr	Integer	Fixed	2	N/A	
Resp	Integer	Fixed	3	N/A	
FwdState	Integer	Fixed	3	N/A	
DBID	Integer	Fixed	8	N/A	
FwdNID	Integer	Derived	7	7/11	FwdNID = NodeID_Width;
FwdTxnID	Integer	Fixed	8	N/A	
DoNotGoToSD	Integer	Fixed	1	N/A	
RetToSrc	Integer	Fixed	1	N/A	
Homenode_ID	Integer	Derived	7	7	Homenode_ID = NodeID_Width;
CCID	Integer	Fixed	2	N/A	
DataID	Integer	Fixed	2	N/A	
BE	Integer	Derived	8	64	BE = wData/8; wData = { 64, 128, 256}
wQos	Integer	Fixed	4	N/A	
wPoison	Integer	Derived	2	4	wPoison = enPoison ? (wData/64) : 0;
wReqflit	Integer	Derived	95	95	wReqflit = wQos + TgtID + SrcID + TxnID + ReturnNID + StashNIDValid + ReturnTxnID + Opcode + Size + wAddr + NS + LikelyShared + AllowRetry + Order + PCrdType + MemAttr + SnpAttr + LPID + Excl + ExCompAck + TraceTag + REQ_RSVDC;
wRspflit	Integer	Derived	34	34	wRspflit = wQos + TgtID + SrcID + TxnID + Opcode + RespErr + Resp + FwdState + DBID + PCrdType + TraceTag;
wDatflit	Integer	Derived	125	125	wDatflit = wQos + TgtID + SrcID + TxnID + Homenode_ID + Opcode + RespErr + Resp + FwdState + DBID + CCID + DataID + TraceTag + DAT_RSVDC + BE + wPoison + wData;
wSnpflit	Integer	Derived	70	70	wSnpflit = wQos + SrcID + TxnID + FwdNID + FwdTxnID + Opcode + wAddr + NS + DoNotGoToSD + RetToSrc + TraceTag - 3;

Commented [BM6]: This can't be correct

23.7. CHI_E Interface

1293 TABLE 23-7: CHI_E INTERFACE DERIVED PARAMETERS-1

Parameter Name	Type		Default	Min/ Max	Description/Derivation
SrcID	Integer	Derived	7	7/11	SrcID = NodeID_Width;
TgtID	Integer	Derived	7	7/11	TgtID = NodeID_Width;
TxnID	Integer	Fixed	12	N/A	
ReturnNID	Integer	Derived	7	7/11	ReturnNID = NodeID_Width;
StashNIDValid	Integer	Fixed	1	N/A	
ReturnTxnID	Integer	Fixed	12	N/A	
REQ_Opcode	Integer	Fixed	7	N/A	
RSP_Opcode	Integer	Fixed	5	N/A	
SNP_Opcode	Integer	Fixed	5	N/A	
DAT_Opcode	Integer	Fixed	4	N/A	
Size	Integer	Fixed	3	N/A	
wAddr	Integer	Derived	48	44/52	Physical address width wAddr = {44, 48, 52}. Obtained from suer input.
NS	Integer	Fixed	1	N/A	
LikelyShared	Integer	Fixed	1	N/A	
AllowRetry	Integer	Fixed	1	N/A	
Order	Integer	Fixed	2	N/A	
PCrdType	Integer	Fixed	4	N/A	
MemAttr	Integer	Fixed	4	N/A	
SnpAttr	Integer	Fixed	1	N/A	
GroupExtID	Integer	Fixed	3	N/A	
LPID	Integer	Fixed	5	N/A	
Excl	Integer	Fixed	1	N/A	
ExCompAck	Integer	Fixed	1	N/A	
TraceTag	Integer	Fixed	1	N/A	
DAT_RSVDC	Integer	Fixed	0	N/A	Not supported and is always fixed at zero
RespErr	Integer	Fixed	2	N/A	
Resp	Integer	Fixed	3	N/A	

Parameter Name	Type		Default	Min/ Max	Description/Derivation
FwdState	Integer	Fixed	3	N/A	
DBID	Integer	Fixed	12	N/A	
FwdNID	Integer	Derived	7	7/11	FwdNID = NodeID_Width;
FwdTxnID	Integer	Fixed	12	N/A	
DoNotGoToSD	Integer	Fixed	1	N/A	
RetToSrc	Integer	Fixed	1	N/A	
Homenode_ID	Integer	Derived	7	7/11	Homenode_ID = NodeID_Width;
CCID	Integer	Fixed	2	N/A	
DataID	Integer	Fixed	2	N/A	
BE	Integer	Derived	8	8/64	BE = wData/8; wData = { 64, 128, 256}
wQos	Integer	Fixed	4	N/A	
wPoison	Integer	Derived	2	4	wPoison = enPoison ? (wData/64) : 0;
wReqflit	Integer	Derived	133	129/181	wReqflit = wQos + TgtID + SrcID + TxnID + ReturnNID + StashNIDValid + ReturnTxnID + Opcode + Size + wAddr + NS + LikelyShared + AllowRetry + Order + PCrdType + MemAttr + SnpAttr + LPID + Excl + ExCompAck + TraceTag + REQ_RSVDC;
wRspflit	Integer	Derived	60	60/68	wRspflit = wQos + TgtID + SrcID + TxnID + Opcode + RespErr + Resp + FwdState + DBID + PCrdType + TraceTag;
wDatflit	Integer	Derived	355[6]	139/431	wDatflit = wQos + TgtID + SrcID + TxnID + Homenode_ID + Opcode + RespErr + Resp + FwdState + DBID + CCID + DataID + TraceTag + DAT_RSVDC + BE + wPoison + wData;
wSnpflit	Integer	Derived	89	85/108	wSnpflit = wQos + SrcID + TxnID + FwdNID + FwdTxnID + Opcode + wAddr + NS + DoNotGoToSD + RetToSrc + TraceTag - 3;

[6] Take the default as 256 bits of data bus (64, 128,256) No poison support, zero bits of DAT_RSVDC

23.8. CXS.B interface

TABLE 23-8 CXS PARAMETERS

Parameter Name	Type		Default	Min/ Max	Description/Derivation
CXS_LAST	Boolean	Fixed	True	N/A	Always set to True per Synopsys UCIe controller requirements.
CXS_MAX_CREDIT	Integer	Derived	15	4/15	Min 4 to allow for full packets to be issued. Default identical to Synopsys UCIe controller
CXS_PROTOCOL_TYPE	Boolean	Fixed	False	N/A	Always set to False since Synopsys UCIe controller does have such a signal
CXSCHECKTYPE	Enum of string	Fixed	None	None or Odd_Byte_ Parity	Currently Synopsys UCIe controller only supports None.
CXSCONTINUOUSDATA	Boolean	Fixed	True	N/A	Always True per Synopsys UCIe controller
CXSDATAFLITWIDTH	Integer	Fixed	512	N/A	Synopsys UCIe controller only supports 512
CXSERRORFULLPKT	Boolean	Fixed	True	N/A	Per Synopsys UCIe controller requirement
CXSMAXPKPERFLIT	Integer	Fixed	2	2	Synopsys UCIe controller actual maximum number of packets per flit is restricted to 1.
CXSLINKCONTROL	String	Fixed	Explicit_Credit_Retu rn	NA	Once it is set as Explicit_Credit_Return, the following CXS links are added: CXSCRDRTN CXSACTIVEREQ CXSACTIVEACK CXSDEACTHINT
wValid	Integer	Fixed	1	N/A	
wData	Integer	Derived	CXSDATAFLITWIDT H	512/512	
wCntl	Integer	Derived	CXSDATAFLITWIDT H =512 : 18 CXSDATAFLITWIDT H =256 14	14/18	In practice is fixed, but will be derived if we ever support a different data width.
wLast	Integer	Fixed	1	N/A	
wPrcltype	Integer	Fixed	3	N/A	
wCrdrtn	Integer	Fixed	1	N/A	
wCrdGnt	Integer	Fixed	1	N/A	
wActiveack	Integer	Fixed	1	N/A	
wActivereq	Integer	Fixed	1	N/A	
wDeacthint	Integer	Fixed	1	N/A	

24. *Single Die variant Derived/Fixed Concerto Parameters*

24.1. System parameter : concertocsmiparams

Table 24-1: concertocsmiparams Parameters

Parameter Name	Type		Default	Min/ Max	Description/Derivation
wTargetId	Integer	Derived			wTargetId = wFUnitId + wFPortId;
wInitiatorId	Integer	Derived			wInitiatorId = wFUnitId + wFPortId;
wMsgId	Integer	Derived			wMsgId = wMessageId;
wAddr	Integer	Derived	0		Derived by mapper code max. of wAddr of all the sockets
wMPF1	Integer	Derived			wMPF1 = max({1+wFUnitId, 1+wMaxChiNodeId, wArgV, wAXIFldSet, wTargetId, wVMIDExt, wFlowId, wInitiatorId, wMsgId});
wMPF2	Integer	Derived			wMPF2 = max({(1+wLPId), (1+wFlowId), wDvmSnpUnqId, wMsgId});
wMPF3	Integer	Derived			wMPF3 = max({wFUnitId, wDvmSnpPartId, wFlowId});
wDId	Integer	Derived			wDId = wFUnitId
nBEPerDW	Integer	Fixed	8		
wBEPerDW	Integer	Fixed	8		

wProtPerDW	Integer	Derived	0	0/8	wProtPerDW = 0;
					if (ResilienceEnable)
					{if TIResiliencyProtectionType == SECDED) {
					wProtPerDW = 8; }
					if (TIResiliencyProtectionType == PARITY) {
					wProtPerDW = 1;
					}
					}

wAuxPerDW	Integer	Fixed	0	0/32	
wDPPerBeat	Integer				Possibly not being used
wDataBitsPerDW	Integer	Fixed	64	64	
wDBadPerDW	Integer	Fixed	1	1	
wDPPerDW	Integer	Derived			wDPPerDW = wDataBitsPerDW + wBEPerDW + wDBadPerDW + wDWId + wProtPerDW + wAuxPerDW;
nSmiVC	Integer	Fixed	1	1	VC are not used in Ncore
wSmiTid	Integer	Derived			wSmiTid = wTargetId;
wSmiSid	Integer	Derived			wSmiSid = wInitaitorId;
wSmiType	Integer	Derived			wSmiType = wCMType;
wSmiMsgId	Integer	Derived			wSmiMsgId = wMsgId;
wSmiUser	Integer	Derived			wSmiUser = wHProt;
wSmiSteer	Integer	Derived		wSmiSteer = wSteering;	wSmiSteer

Parameter Name	Type		Default	Min/ Max	Description/Derivation
wSmiTier	Integer	Derived		wSmiTier = wTTier;	wSmiTier
wSmiQos	Integer	Derived		wSmiQos = wQL;	wSmiQos
wSmiPri	Integer	Derived		wSmiPri = wPriority;	wSmiPri
wSmiNDPLen	Integer	Fixed	8		wSmiNDPLen
wSmiNDP	Integer	Will be derived		This will be defined at port level.	wSmiNDP
wSmiErr	Integer	Fixed	0	Not Used in Ncore	wSmiErr
wSmiRoute	Integer	Fixed	0	Not Used in Ncore	wSmiRoute
wSmiClass	Integer	Fixed	0	Not Used in Ncore	wSmiClass
wSmiSeqnum	Integer	Fixed	0	Not Used in Ncore	wSmiSeqnum
wSmiAddr	Integer	Fixed	0	Not Used in Ncore	wSmiAddr
wSmiLen	Integer	Fixed	0	Not Used in Ncore	wSmiLen
wSmiVNid	Integer	Fixed	0	Not Used in Ncore	wSmiVNid
wSmiProt	Integer	Fixed	0	Not Used in Ncore	wSmiProt
wSmiTxnHdr	Integer	Fixed	0	Not Used in Ncore	wSmiTxnHdr
nSmiDPvc	Integer	Fixed	1		nSmiDPvc
wSmiDPlast	Integer	Fixed	1		wSmiDPlast
wSmiDPdata	Integer	Derived	{64,128,256}	Will be defined at block level wSmiDPdata: ncore3 uses 256 max..	wSmiDPdata
wSmiDPuser	Integer	Fixed	0	Not Used in Ncore	wSmiDPuser
wSmiDPbe	Integer	Fixed	0	Not Used in Ncore	wSmiDPbe
wSmiDPid	Integer	Fixed	0	Not Used in Ncore	wSmiDPid
wSmiDPerr	Integer	Fixed	0	Not Used in Ncore	wSmiDPerr
wSmiDPresp	Integer	Fixed	0	Not Used in Ncore	wSmiDPresp
wSmiDPdummy	Integer	Fixed	0	Not Used in Ncore	wSmiDPdummy

1308

24.2. System parameter : concertocparams

1311 Table 24-2: concertocparams Parameters

Parameter Name	Type	Origin	Default	Min/ Max	Description/Derivation
wCacheLine	Integer	Fixed	6	N/Ax	2^(wCacheLine) is the Cache line width in byte 64B Cache Line
wDWId	Integer	Fixed	3	N/A	Number of Bits Identifing a DW Within a CG
wDBad	Integer	Fixed	1	N/A	Width of the signal Dbad in bits. When set, it indicates that a data DW is corrupted (i.e. Bad) and therefore must not be consumed in a computation
wSysReqOp	Integer	Fixed	4	N/A	
wRequestorId	Integer	Derived	Derived	Derived	wRequestorId = wFUnitId
wValid	Integer	Fixed	1	N/A	
wReady	Integer	Fixed	1	N/A	
wLast	Integer	Fixed	1	N/A	
wStashFUnitId	Integer	Derived	Derived	Derived	wStashFUnitId = wFUnitId;
wStashNId	Integer	Derived	Derived	Derived	wStashNId = wStashFUnitId;
HProtEnable	Boolean	Fixed	False	N/A	HProt is being defined…?
TTierEnable	Boolean	Fixed	False	N/A	Not used in Ncore 3.x
QLEnable	Boolean	Fixed	False	N/A	Not used in Ncore 3.x
SteeringEnable	Boolean	Fixed	False	N/A	Not used in Ncore 3.x
PriorityEnable	Boolean	Fixed	True	N/A	Not used by the design
wTargetId	Integer	Derived	Derived	Derived	wTargetId = wFUnitId + wFPortId; (from Common)
wInitiatorId	Integer	Derived	Derived	Derived	wInitiatorId = wFUnitId + FPortId; (from Common)
wCMType	Integer	Fixed	8	N/A	Width of the concerto cm_type field.
wMessageId	Integer	Derived	Derived	Derived	wMessageId = max({log2MaxAiuCredits, log2MaxDceCredits, log2MaxDmiCredits, log2MaxDiiCredits}); log2MaxAiuCredits = 1 + log2ceil(Max(1,maxOttCtrlEntries, maxSttCtrlEntries)) log2MaxDceCredits = log2ceil(max(1,maxAttCtrlEntries)) log2MaxDmiCredits = log2ceil(max(1, maxDmiWttCtrlEntries,maxDmiRttCtrlEntries)) log2MaxDIICredits = log2ceil(max(1, maxDiiWttCtrlEntries,maxDiiRttCtrlEntries))

Commented [BM7]: Check this one out.

Parameter Name	Type	Origin	Default	Min/ Max	Description/Derivation

wHProt	Integer	Derived	0	0/12	if (! ResilienceEnable) { wHProt = Integer(0); } else
					{
					  if (TIResilienceProtectionType == NONE) {
					  wHProt = 0;
					} else
					  if (TIResilienceProtectionType == PARITY) {
					  wHProt = 1;
					} else {
					  auto temp = wTargetId + wInitiatorId
					  + wCMType + wMessageId;
					  int64_t ecc_width = 3;
					  while ($2^{(ecc\_width - 1)}$ < (temp + ecc_width)) {
					  ecc_width += 1;
					  }
					  wHProt = ecc_width;
					  }
					}

wTTier	Integer	Fixed	0	N/A	Not used by Ncore
wSteering	Integer	Fixed	0	N/A	Not used by Ncore
wPriority	Integer	Derived	Derived	Derived	wPriority = useQos ? 3 : 0;
wQL	Integer	Fixed	0	N/A	Not used by Ncore
wCMHeader	Integer	Derived	Derived		wCMHeader = wTargetId + wInitiatorId + wCMType + wMessageId + wHProt + wTTier + wSteering + wPriority + wQL;
wCMStatus	Integer	Fixed	8	N/A	
wVZ	Integer	Fixed	1	N/A	
wCA	Integer	Fixed	1	N/A	
wAC	Integer	Fixed	1	N/A	
wCH	Integer	Fixed	1	N/A	
wST	Integer	Fixed	1	N/A	
wEN	Integer	Fixed	1	N/A	
wES	Integer	Fixed	1	N/A	
wNS	Integer	Fixed	1	N/A	
wPR	Integer	Fixed	1	N/A	
wOR	Integer	Fixed	2	N/A	
wLK	Integer	Fixed	2	N/A	
wRL	Integer	Fixed	2	N/A	
wTM	Integer	Fixed	1	N/A	
wUP	Integer	Fixed	2	N/A	
wPrimary	Integer	Fixed	1	N/A	
wMW	Integer	Fixed	1	N/A	
wEO	Integer	Fixed	0	N/A	
wSize	Integer	Fixed	3	N/A	
wIntfSize	Integer	Fixed	2	N/A	
wTOF	Integer	Fixed	3	N/A	
wQoS	Integer	Derived	Derived	0 or 4	wQos = useQos ? 4 : 0;

Commented [BM8]: review

Parameter Name	Type	Origin	Default	Min/ Max	Description/Derivation
wTNType	Integer	Fixed	8	N/A	
wAddr	Integer	Derived	Derived	Derived	From NC_ConcertoCSMIParams.json
wMPF1	Integer	Derived	Derived	Derived	From NC_ConcertoCSMIParams.json
wMPF2	Integer	Derived	Derived	Derived	From NC_ConcertoCSMIParams.json
wMPF3	Integer	Derived	Derived	Derived	From NC_ConcertoCSMIParams.json
wDId	Integer	Derived	Derived	Derived	From NC_ConcertoCSMIParams.json
wRBGen	Integer	Fixed	1	N/A	
wRBID	Integer	Derived			From NC_NcoreCredits.json Max(1, wDiiRb, wDiiWtt, wDmiDceRb, wDveRb)+ wRBGen wDiiRb = log2ceil(max(1,maxDiiWttCtrlEntries)) wDiiWtt = log2ceil(max(1,maxDiiRbCredits)) wDmiDceRb = log2ceil(max(1,nTotalDceRbCredits + maxDmiRbCredits)) wDveRb = log2ceil(max(1,nSkidEntries)) wRBGen = 1
wRType	Integer	Fixed	1	N/A	
wNdpAux	Integer	Derived		0/32	Derivation is in mapping code = max {ArUser, AwUser}
wNdpProt	Integer				Is it being used?
wRMessageId	Integer			0/12	wRMessageId = wMessageId;
wTNMsg	Integer			0/16	
ECMType	Integer				Is it being used? If there is no default vaule, Maestro is set it as 0
wArgV	Integer		6	3/8	MAES-3383. Default is changed from 3 to 6.
wFlowId	Integer	Derived	5	5/20	Derivation is in mapping code: max {ArId, AwId}
wLPId	Integer		0	0/5	Derivation is in mapping code • ACE: Determined as log2 (number of processors in the largest cluster) CHI_B: 5

1314

24.3. System parameter : ConcertoCRequestMessagefields

1317 Table 24-3: ConcertoCRequestMessageFields Parameters

Parameter Name	Type	Description/Derivation
wCMDNdp	Integer	wCMDNdp = wCMStatus + wVZ+wCA + wAC + wCH + wST + wEN + wES + wNS + wPR + wOR + wLK + wRL + wTM + wSize + wIntfSize + wTOF + wQoS + wAddr + wMPF1 + wMPF2 + wDId+ wNdpAux + wCMDMProt;
wSYSNdp	Integer	wSYSNdp = wCMStatus + wSysReqOp + wRMessageId + wTM + wSYSMProt + wRequestorId;
wSNPNdp	Integer	wSNPNdp = wCMStatus + wVZ + wCA + wAC + wNS + wPR + wRL + wTM+wUP+wIntfSize + wTOF+ wQoS+ wAddr+ wMPF1 + wMPF2 + wMPF3+ wDId+ wRBID+ wNdpAux+ wSNPMProt;
wMRDNdp	Integer	wMRDNdp = wCMStatus + wAC + wNS+ wPR+ wRL+ wTM + wSize + wIntfSize+ wQoS+ wAddr+ wMPF1 + wMPF2 + wNdpAux + wMRDMProt;
wUPDNdp	Integer	wUPDNdp = wCMStatus + wNS + wAddr + wUPDMProt + wQos + wTM;
wHNTNdp	Integer	this transaction type will not implemented in Ncore 3.x
wSTRNdp	Integer	wSTRNdp = wCMStatus+ wMPF1 + wMPF2 + wRBID + wRMessageId + wIntfSize + wTM + wSTRMProt;
wTUNNdp	Integer	
wRBRNdp	Integer	wRBRNdp = wCMStatus + wVZ + wCA + wAC + wNS + wPR+ wRL+ wMW + wSize+ wTOF+ wQoS + wAddr + wMPF1+ wRType+ wRBID + wRBRMProt+ wNdpAux + wTM;
wRBUNdp	Integer	wRBUNdp = wCMStatus + wRL + wRBID + wTM + wRBUMProt
wDTRNdp	Integer	wDTRNdp = wCMStatus + wRL + wTM + wMPF1 + wNdpAux + wRMessageId + wDTRMProt;
wDTWNdp	Integer	wDTWNdp = wCMStatus+ wRL+ wTM + wPrimary + wMPF1 + wMPF2 + wRBID+ wNdpAux + wDTWMProt + wIntfSize;
wDTWDBGNdp	Integer	wDTWDBGNdp = wCMSTatus + wRT + wTM + wNdpAux + wDTWDBGMProt

wCMDMProt	Integer	if (! ResilienceEnable) {wCMDMProt = 0; }
		else {
		if (TIResiliencyProtectionType == NONE)
		{wCMDMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY)
		{wCMDMProt = 1;
		} else {
		auto temp = wCMStatus+ wVZ + wCA + wAC+ wCH"+ wST + wEN
		+ wES+ wNS + wPR + wOR + wLK + wRL + wTM + wSize
		+ wIntfSize + wTOF + wQoS + wAddr + wMPF1 + wMPF2
		+ wDId + wNdpAux;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wCMDMProt = ecc_width;
		}
		}

Parameter Name	Type	Description/Derivation

wSYSMProt	Integer	if (! ResilienceEnable) {wSYSMProt = 0; }
		else {
		  if (TIResiliencyProtectionType == NONE) {
		  wSYSMProt = 0;
		  } else if (TIResiliencyProtectionType == PARITY) {
		  wSYSMProt = 1;
		  } else {
		  auto temp = wCMStatus + wSysReqOp + wRMessageId + wTM + wRequestorId;
		  int64_t ecc_width = 3;
		  while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		  ecc_width += 1;
		  }
		  wSYSMProt = ecc_width;
		  }
		}

wSNPMProt	Integer	if (! ResilienceEnable) {wSNPMProt = 0;}
		else {
		  if (TIResiliencyProtectionType == NONE) {
		  wSNPMProt = 0;
		  } else if (TIResiliencyProtectionType == PARITY) {
		  wSNPMProt = 1;
		  } else {
		  auto temp = wCMStatus + wVZ + wCA + wAC + wNS + wPR + wRL
		  + wTM + wUP + wIntfSize + wTOF + wQoS + wAddr
		  + wMPF1+ wMPF2 + wMPF3 + wDId+ wRBID + wNdpAux;
		  int64_t ecc_width = 3;
		  while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		  ecc_width += 1;
		  }
		  wSNPMProt = ecc_width;
		  }
		}

wMRDMProt	Integer	if (! ResilienceEnable) {wMRDMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		  wMRDMProt = 0;
		  } else if (TIResiliencyProtectionType == PARITY) {
		  wMRDMProt = 1;
		  } else {
		  auto temp = wCMStatus+ wAC+ wNS+ wPR+ wRL+ wTM + wSize
		  + wIntfSize+ wQoS+ wAddr+ wMPF1+ wMPF2 + wNdpAux
		  int64_t ecc_width = 3;
		  while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		  ecc_width += 1;
		  }
		  wMRDMProt = ecc_width;
		  }
		}

wHNTMProt	Integer	
wTUNMProt	Integer	

Parameter Name	Type	Description/Derivation

wUPDMProt	Integer	if (! ResilienceEnable) {wUPDMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wUPDMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wUPDMProt = 1;
		} else {
		auto temp = wCMStatus + wNS+ wAddr+ wQos + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wUPDMProt = ecc_width;
		}
		}

wSTRMProt	Integer	if (! ResilienceEnable) {wSTRMProt = 0;}
		else {if (TIResiliencyProtectionType == NONE) {
		wSTRMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wSTRMProt = 1;
		} else {
		auto temp = wCMStatus + wMPF1 + wMPF2 + wRBID + wRMessageId
		+ wIntfSize + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wSTRMProt = ecc_width;
		}
		}

wRBRMProt	Integer	if (! ResilienceEnable) {wRBRMProt = 0; }
		else { if (TIResiliencyProtectionType == NONE) {
		wRBRMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wRBRMProt = 1;
		} else {
		auto temp = wCMStatus + wVZ + wCA + wAC + wNS + wPR + wRL
		+ wMW + wSize+ wTOF + wQoS + wAddr + wMPF1
		+ wRType + wRBID + wNdpAux + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wRBRMProt = ecc_width;
		}
		}

wRBUMProt	Integer	if (! ResilienceEnable) {wRBUMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wRBUMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wRBUMProt = 1;
		} else {
		auto temp = wCMStatus + wRL + wRBID + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) <)temp + ecc_width)) {
		ecc_width += 1;
		}
		wRBUMProt = ecc_width;
		}
		}

Parameter Name	Type	Description/Derivation

wDTRMProt	Integer	if (! ResilienceEnable) { wDTRMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wDTRMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wDTRMProt = 1;
		} else {
		auto temp = wCMStatus + wRL + wTM + wMPF1 + wNdpAux
		+ wRMessageId;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wDTRMProt = ecc_width;
		}
		}

wDTWMProt	Integer	if (! ResilienceEnable) {wDTWMProt = 0; }
		else { if (TIResiliencyProtectionType == NONE) {
		wDTWMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wDTWMProt = 1;
		} else {
		auto temp = wCMStatus + wRL + wTM + wPrimary + wMPF1 + wMPF2
		+ wRBID + wNdpAux + wIntfSize ;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wDTWMProt = ecc_width;
		}
		}

wDTWDBGMProt	Integer	if (! ResilienceEnable) {wDTWDBGMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wDTWDBGMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wDTWDBGMProt = 1;
		} else {
		auto temp = wCMStatus + wRL + wTM + wPrimary + wMPF1+ wMPF
		+ wRBID + wNdpAux+ wIntfSize;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wDTWDBGMProt = ecc_width;
		}
		}

1320

24.4. System parameter : ConcertoCResponseMessageFields

1323 Table 24-4: ConcertoCResponseMessageFields Parameters

Parameter Name	Type	Description/Derivation
wCCMDrspNdp	Integer	wCCMDrspNdp = wCMStatus + wRMessageId + wTM + wCCMDrspMProt
wSYSrspNdp	Integer	wSYSrspNdp = wCMStatus + wRMessageId + wTM + wSYSrspMProt
wNCCMDrspNdp	Integer	wNCCMDrspNdp = wCMStatus+ wRMessageId + wTM+ wNCCMDrspMProt
wSNPrspNdp	Integer	wSNPrspNdp = wCMStatus + wIntfSize + wMPF1 + wRMessageId + wTM + wSNPrspMProt
wDTWrspNdp	Integer	wDTWrspNdp = wCMStatus + wRMessageId + wRL + wTM + wDTWrspMProt
wDTWDBGrspNdp	Integer	wDTWDBGrspNdp = wCMStatus + wRMessageId + wRL + wTM + wDTWDBGrspMProt
wDTRrspNdp	Integer	wDTRrspNdp = wCMStatus + wRMessageId + wTM + wDTRrspMProt
wHNTrspNdp	Integer	Not used in Ncore 3
wMRDrspNdp	Integer	wMRDrspNdp = wCMStatus + wRMessageId + wTM + wMRDrspMProt
wSTRrspNdp	Integer	wSTRrspNdp = wCMStatus + wRMessageId + wTM + wSTRrspMProt;
wUPDrspNdp	Integer	wUPDrspNdp = wCMStatus+ wRMessageId+ wTM + wUPDrspMProt
wRBRrspNdp	Integer	wRBRrspNdp = wCMStatus + wRMessageId + wTM + wRBRrspMProt
wRBUrspNdp	Integer	wRBUrspNdp = wCMStatus + wRMessageId + wTM+ wRBUrspMProt
wCMPrspNdp	Integer	wCMPrspNdp = wCMStatus+ wRMessageId + wTM + wCMPrspMProt
wCMErspNdp	Integer	Not Used in Ncore3
wTUNrspNdp	Integer	Not Used in Ncore3
wTRErspNdp	Integer	Not Used in Ncore3

wCCMDrspMProt	Integer	if (! ResilienceEnable) { wCCMDrspMProt = 0; }
		else {
		if (TIResiliencyProtectionType == NONE) {
		wCCMDrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wCCMDrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wCCMDrspMProt = ecc_width;
		}
		}

wSYSrspMProt	Integer	if (! ResilienceEnable) {wSYSrspMProt = 0; }
		else {
		if (TIResiliencyProtectionType == NONE) {
		wSYSrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wSYSrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId +wTM
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;

		} wSYSrspMProt = ecc_width } }

wNCCMDrspMProt	Integer	if (! ResilienceEnable) {wNCCMDrspMProt = 0; }
		else {
		if (TIResiliencyProtectionType == NONE) {
		wNCCMDrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wNCCMDrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wNCCMDrspMProt = ecc_width;
		}
		}

wSNPrspMProt	Integer	if (! ResilienceEnable) {wSNPrspMProt = 0; }
		else {
		if (TIResiliencyProtectionType == NONE) {
		wSNPrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wSNPrspMProt = 1;
		} else {
		auto temp = wCMStatus + wIntfSize + wMPF1 + wRMessageId +wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;",
		}
		wSNPrspMProt = ecc_width;
		}
		}

wDTWrspMProt	Integer	if (! ResilienceEnable) {wDTWrspMProt = 0; }
		else {
		if (TIResiliencyProtectionType == NONE) {
		wDTWrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wDTWrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wRL + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;",
		};
		wDTWrspMProt = ecc_width;
		}
		}

wDTWDBGrspMProt	Integer	if (! ResilienceEnable) {wDTWDBGrspMProt = 0; }
		else { if (TIResiliencyProtectionType == NONE) {
		wDTWDBGrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wDTWDBGrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wRL + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < temp + ecc_width) {
		ecc_width += 1;
		}
		wDTWDBGrspMProt = ecc_width;

		} }

wDTRrspMProt	Integer	if (! ResilienceEnable) {wDTRrspMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wDTRrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wDTRrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wDTRrspMProt = ecc_width;
		}
		}

wHNTrspMProt	Integer	

wMRDrspMProt	Integer	if (! ResilienceEnable) {wMRDrspMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wMRDrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wMRDrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wMRDrspMProt = ecc_width;
		}
		}

wSTRrspMProt	Integer	if (! ResilienceEnable) {wSTRrspMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wSTRrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wSTRrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId +wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wSTRrspMProt = ecc_width;
		}
		}

wUPDrspMProt	Integer	if (! ResilienceEnable) {wUPDrspMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wUPDrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wUPDrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wUPDrspMProt = ecc_width;
		}
		}

wRBRrspMProt	Integer	if (! ResilienceEnable) {wRBRrspMProt = 0; } else {if (TIResiliencyProtectionType == NONE) {

		wRBRrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wRBRrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId +wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wRBRrspMProt = ecc_width
		}
		}

wRBUrspMProt	Integer	if (! ResilienceEnable) {wRBUrspMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wRBUrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wRBUrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId +wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < temp + ecc_width) {
		ecc_width += 1;
		}
		wRBUrspMProt = ecc_width;
		}
		}

wCMPrspMProt	Integer	if (! ResilienceEnable) {wCMPrspMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE)
		{wCMPrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY)
		{wCMPrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < temp + ecc_width) {
		ecc_width += 1;
		}
		wCMPrspMProt = ecc_width;
		}
		}

25. *Multi-Die Variant of Derived/Fixed Concerto Parameters*

25.1. System parameter : concertocsmiparams (multi-die)

TABLE 25-1: CONCERTOCSMIPARAM PARAMETERS FOR MULTI-DIE CONFIGURATION

Parameter Name	Type		Default	Min/ Max	Description/Derivation
wTargetId	Integer	Derived			wTargetId = wFUnitId + wFPortId + wChipletId + wLinkId. All the components are now fixed and should be 13.
wInitiatorId	Integer	Derived			wInitiatorId = wFUnitId + wFPortId + wChipletId + wLinkId. All the components are now fixed and should be 13.
wMsgId	Integer	Derived	10	NA	wMsgId = wMessageId
wAddr	Integer	Fixed	52	NA	Fixed to be 52
wMPF1	Integer	Fixed	21	NA	Now fixed. Check: wMPF1 = max({1+wFUnitId, 1+wMaxChiNodeId, wArgV, wAXIFIdSet, wTargetId, wVMIDExt, wFlowId, wInitiatorId, wMsgId});
wMPF2	Integer	Fixed	21	NA	Now fixed. Check wMPF2 = max({{4+wLPId), (1+wFlowId), wDvmSnpUnqId, wMsgId});
wMPF3	Integer	Fixed	21	NA	Now fixed. Check wMPF3 = max({wFUnitId+wChipletId, wDvmSnpPartId, wFlowId});
wDId	Integer	Derived			wDId = wFUnitId + wChipletId, Did is extended to contain the Global FUnitId
nBEPerDW	Integer	Fixed	8	NA	Unchanged
wBEPerDW	Integer	Fixed	8	NA	Unchanged

wProtPerDW	Integer	Derived	0	0/8	wProtPerDW = 0;
					if (ResilienceEnable)
					{if TIResiliencyProtectionType == SECDED) {
					wProtPerDW = 8; }
					if (TIResiliencyProtectionType == PARITY) {
					wProtPerDW = 1;
					}
					}

wAuxPerDW	Integer	Fixed	0	NA	
wDPPerBeat	Integer				Possibly not being used
wDataBitsPerDW	Integer	Fixed	64	64	
wDBadPerDW	Integer	Fixed	1	1	
wDPPerDW	Integer	Derived			wDPPerDW = wDataBitsPerDW + wBEPerDW + wDBadPerDW + wDWId + wProtPerDW + wAuxPerDW;
nSmiVC	Integer	Fixed	1	1	VC are not used in Ncore
wSmiTid	Integer	Derived			wSmiTid = wTargetId;
wSmiSid	Integer	Derived			wSmiSid = wInitaitorId;
wSmiType	Integer	Derived			wSmiType = wCMType;
wSmiMsgId	Integer	Derived			wSmiMsgId = wMsgId;
wSmiUser	Integer	Derived			wSmiUser = wHProt;
wSmiSteer	Integer	Derived			wSmiSteer = wSteering;
wSmiTier	Integer	Derived			wSmiTier = wTTier;

Parameter Name	Type		Default	Min/ Max	Description/Derivation
wSmiQos	Integer	Derived			wSmiQos = wQL;
wSmiPri	Integer	Derived			wSmiPri = wPriority;
wSmiNDPLen	Integer	Fixed	8	8	
wSmiNDP	Integer	Will be derived			This will be defined at port level.
wSmiErr	Integer	Fixed	1	1	
wSmiRoute	Integer	Fixed	0	0	Unchanged
wSmiClass	Integer	Fixed	0	0	Unchanged
wSmiSeqnum	Integer	Fixed	0	0	Unchanged
wSmiAddr	Integer	Fixed	0	0	Unchanged
wSmiLen	Integer	Fixed	0	0	Unchanged
wSmiVNid	Integer	Fixed	0	0	Unchanged
wSmiProt	Integer	Fixed	0	0	Unchanged
wSmiTxnHdr	Integer	Fixed	0	0	Unchanged
nSmiDPvc	Integer	Fixed	1	1	Unchanged
wSmiDPlast	Integer	Fixed	1	1	Unchanged
wSmiDPdata	Integer	Derived		{64,128, 256,512}	Will be defined at block level wSmiDPdata: ncore3 uses 256-bit wide maximum. 512-bit wide is not verified. And 512-bit wide will be used strictly for GIU and the switches it connects to.
wSmiDPuser	Integer	Fixed	0	0	Unchanged
wSmiDPbe	Integer	Fixed	0	0	Unchanged
wSmiDPid	Integer	Fixed	0	0	Unchanged
wSmiDPerr	Integer	Fixed	0	0	Unchanged
wSmiDPresp	Integer	Fixed	0	0	Unchanged
wSmiDPdummy	Integer	Fixed	0	0	Unchanged

25.2. System parameter : ConcertoCparams

TABLE 25-2 CONCERTOCPARAMS PARAMETERS FOR MULTI-DIE VARIANT

Parameter Name	Type	Origin	Default	Min/ Max	Description/Derivation
wCacheLine	Integer	Fixed	6	N/A	Cache line width in byte 64B Cache Line
wDWId	Integer	Fixed	3	N/A	Number of Bits Identifying a DW Within a CG
wDBad	Integer	Fixed	1	N/A	Width of the signal DBad in bits. When set, it indicates that a data DW is corrupted (i.e. Bad) and therefore must not be consumed in a computation
wSysReqOp	Integer	Fixed	4	N/A	
wRequestorId	Integer	Derived	Derived	Derived	wRequestorId = wFunitId + wChipletId
wValid	Integer	Fixed	1	N/A	
wReady	Integer	Fixed	1	NA	
wLast	Integer	Fixed	1	N/A	
wStashFUnitId	Integer	Derived	Derived	Derived	wStashFUnitId = wFUnitId; Uchanged, not stash allowed across chiplets.
wStashNId	Integer	Derived	Derived	Derived	wStashNId = wStashFUnitId; Unchanged, not stash allowed across chiplets.
HProtEnable	Boolean	Fixed	False	N/A	HProt is being defined…?
TTierEnable	Boolean	Fixed	False	N/A	Not used in Ncore 3.x Unchanged
QLEnable	Boolean	Fixed	False	N/A	Not used in Ncore 3.x Unchanged
SteeringEnable	Boolean	Fixed	False	N/A	Not used in Ncore 3.x Unchanged
PriorityEnable	Boolean	Fixed	True	N/A	
wTargetId	Integer	Derived			wTargetId = wFUnitId + wFPortId + wChipletId + wLinkId; (from Common)
wInitiatorId	Integer	Derived			wInitiatorId = wFUnitId + wFPortId + wChipletId + wLinkId; (from Common)
wCMType	Integer	Fixed	8	8/8	Unchanged
wMessageId	Integer	Fixed	10	NA	Max outstanding of 1028.

wHProt	Integer	Derived	Derived	Derived	if (! ResilienceEnable) { wHProt = Integer(0); } else {
					if (TIResilienceProtectionType == NONE) {
					wHProt = 0;
					} else
					if (TIResilienceProtectionType == PARITY) {
					wHProt = 1;
					} else {
					auto temp = wTargetId + wInitiatorId
					+ wCMType + wMessageId;
					int64_t ecc_width = 3;
					while (2(ecc_width - 1) < (temp + ecc_width)) {
					ecc_width += 1;
					}
					wHProt = ecc_width;
					}
					}

Commented [BM9]: To Review

Parameter Name	Type	Origin	Default	Min/ Max	Description/Derivation
wTTier	Integer	Fixed	0	0	Not used (unchanged)
wSteering	Integer	Fixed	0	0	Not used (unchanged)
wPriority	Integer	Fixed	3	3	Now fixed to be always present.
wQL	Integer	Fixed	0	NA	
wCMHeader	Integer	Derived			wCMHeader = wTargetId + wInitiatorId + wCMType + wMessageId + wHProt + wTTier + wSteering + wPriority + wQL;
wCMStatus	Integer	Fixed	8	8	Unchanged
wVZ	Integer	Fixed	1	1	Unchanged
wCA	Integer	Fixed	1	1	Unchanged
wAC	Integer	Fixed	1	1	Unchanged
wCH	Integer	Fixed	1	1	Unchanged
wST	Integer	Fixed	1	1	Unchanged
wEN	Integer	Fixed	1	1	Unchanged
wES	Integer	Fixed	1	1	Unchanged
wNS	Integer	Fixed	1	1	Unchanged
wPR	Integer	Fixed	1	1	Unchanged
wOR	Integer	Fixed	2	2	Unchanged
wLK	Integer	Fixed	2	2	Unchanged
wRL	Integer	Fixed	2	2	Unchanged
wTM	Integer	Fixed	1	1	Unchanged
wUP	Integer	Fixed	2	2	Unchanged
wPrimary	Integer	Fixed	1	1	Unchanged
wMW	Integer	Fixed	1	1	Unchanged
wEO	Integer	Fixed	0		Unchanged
wSize	Integer	Fixed	3	3/4	Unchanged
wIntfSize	Integer	Fixed	2	2/3	Unchanged
wTOF	Integer	Fixed	3	NA	Unchanged
wQoS	Integer	Fixed	4	NA	Now always assumes QOS to be present.
wTNType	Integer	Fixed	8	NA	Unchanged
wAddr	Integer	Derived			From NC_ConcertoCSMIParams.json (will be fixed to 52 there)
wMPF1	Integer	Derived			From NC_ConcertoCSMIParams.json (will be fixed to 21 there)
wMPF2	Integer	Derived			From NC_ConcertoCSMIParams.json (will be fixed to 21 there)
wMPF3	Integer	Derived			From NC_ConcertoCSMIParams.json (will be fixed 21 there)
wDId	Integer	Derived			From NC_ConcertoCSMIParams.json
wRBID	Integer	Derived			From NC_ConcertoCSMIParams.json (will be fixed there)
wRType	Integer	Fixed	1	1	
wNdpAux	Integer	Derived	32	32	Always 32.
wNdpProt	Integer				Is it being used?
wRMessageId	Integer			0/12	wRMessageId = wMessageId;

Parameter Name	Type	Origin	Default	Min/ Max	Description/Derivation
wTNMsg	Integer	Fixed	0	NA	Not used
ECMType	Integer	Fixed	0	NA	Not used
wArgV	Integer		6	3/8	MAES-3383. Default is changed from 3 to 6.
wFlowId	Integer	Derived	5	5/20	Derivation is in mapping code: max {ArId, AwId}
wLPId	Integer		0	0/5	Derivation is in mapping code ACE: Determined as log2 (number of processors in the largest cluster) CHI_B: 5

25.3. System parameter : ConcertoCRequestMessagefields

1338

TABLE 25-3: CONCERTOCREQUESTMESSAGEFIELDS PARAMETERS

Parameter Name	Type	Description/Derivation
wCMDNdp	Integer	wCMDNdp = wCMStatus + wVZ+wCA + wAC + wCH + wST + wEN + wES + wNS + wPR + wOR + wLK + wRL + wTM + wSize + wIntfSize + wTOF + wQoS + wAddr + wMPF1 + wMPF2 + wDId+ wNdpAux + wCMDMProt;
wSYSNdp	Integer	wSYSNdp = wCMStatus + wSysReqOp + wRMessageId + wTM + wSYSMProt + wRequestorId;
wSNPNdp	Integer	wSNPNdp = wCMStatus + wVZ + wCA + wAC + wNS + wPR + wRL + wTM+wUP+wIntfSize + wTOF+ wQoS+ wAddr+ wMPF1 + wMPF2 + wMPF3+ wDId+ wRBID+ wNdpAux+ wSNPMProt;
wMRDNdp	Integer	wMRDNdp = wCMStatus + wAC + wNS+ wPR+ wRL+ wTM + wSize + wIntfSize+ wQoS+ wAddr+ wMPF1 + wMPF2 + wNdpAux + wMRDMProt;
wUPDNdp	Integer	wUPDNdp = wCMStatus + wNS + wAddr + wUPDMProt + wQos + wTM;
wHNTNdp	Integer	this transaction type will not implemented in Ncore 3.x
wSTRNdp	Integer	wSTRNdp = wCMStatus+ wMPF1 + wMPF2 + wRBID + wRMessageId + wIntfSize + wTM + wSTRMProt;
wTUNNdp	Integer	
wRBRNdp	Integer	wRBRNdp = wCMStatus + wVZ + wCA + wAC + wNS + wPR+ wRL+ wMW + wSize+ wTOF+ wQoS + wAddr + wMPF1+ wRType+ wRBID + wRBRMProt+ wNdpAux + wTM;
wRBUNdp	Integer	wRBUNdp = wCMStatus + wRL + wRBID + wTM + wRBUMProt
wDTRNdp	Integer	wDTRNdp = wCMStatus + wRL + wTM + wMPF1 + wNdpAux + wRMessageId + wDTRMProt;
wDTWNdp	Integer	wDTWNdp = wCMStatus+ wRL+ wTM + wPrimary + wMPF1 + wMPF2 + wRBID+ wNdpAux + wDTWMProt + wIntfSize;
wDTWDBGNdp	Integer	wDTWDBGNdp = wCMSTatus + wRT + wTM + wNdpAux + wDTWDBGMProt

Parameter Name	Type	Description/Derivation

wCMDMProt	Integer	if (! ResilienceEnable) {wCMDMProt = 0; }
		else {
		if (TIResiliencyProtectionType == NONE)
		{wCMDMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY)
		{wCMDMProt = 1;
		} else {
		auto temp = wCMStatus+ wVZ + wCA + wAC+ wCH"+ wST + wEN
		+ wES+ wNS + wPR + wOR + wLK + wRL + wTM + wSize
		+ wIntfSize + wTOF + wQoS + wAddr + wMPF1 + wMPF2
		+ wDId + wNdpAux;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wCMDMProt = ecc_width;
		}
		}

wSYSMProt	Integer	if (! ResilienceEnable) {wSYSMProt = 0; }
		else {
		if (TIResiliencyProtectionType == NONE) {
		wSYSMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wSYSMProt = 1;
		} else {
		auto temp = wCMStatus + wSysReqOp + wRMessageId + wTM + wRequestorId;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wSYSMProt = ecc_width;
		}
		}

wSNPMProt	Integer	if (! ResilienceEnable) {wSNPMProt = 0;}
		else {
		if (TIResiliencyProtectionType == NONE) {
		wSNPMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wSNPMProt = 1;
		} else {
		auto temp = wCMStatus + wVZ + wCA + wAC + wNS + wPR + wRL
		+ wTM + wUP + wIntfSize + wTOF + wQoS + wAddr
		+ wMPF1+ wMPF2 + wMPF3 + wDId+ wRBID + wNdpAux;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wSNPMProt = ecc_width;
		}
		}

Parameter Name	Type	Description/Derivation

wMRDMProt	Integer	if (! ResilienceEnable) {wMRDMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wMRDMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wMRDMProt = 1;
		} else {
		auto temp = wCMStatus+ wAC+ wNS+ wPR+ wRL+ wTM + wSize
		+ wIntfSize+ wQoS+ wAddr+ wMPF1+ wMPF2 + wNdpAux
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wMRDMProt = ecc_width;
		}
		}

wHNTMProt	Integer	
wTUNMProt	Integer	

wUPDMProt	Integer	if (! ResilienceEnable) {wUPDMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wUPDMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wUPDMProt = 1;
		} else {
		auto temp = wCMStatus + wNS+ wAddr+ wQos + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wUPDMProt = ecc_width;
		}
		}

wSTRMProt	Integer	if (! ResilienceEnable) {wSTRMProt = 0;}
		else {if (TIResiliencyProtectionType == NONE) {
		wSTRMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wSTRMProt = 1;
		} else {
		auto temp = wCMStatus + wMPF1 + wMPF2 + wRBID + wRMessageId
		+ wIntfSize + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wSTRMProt = ecc_width;
		}
		}

Parameter Name	Type	Description/Derivation

wRBRMProt	Integer	if (! ResilienceEnable) {wRBRMProt = 0; }
		else { if (TIResiliencyProtectionType == NONE) {
		wRBRMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wRBRMProt = 1;
		} else {
		auto temp = wCMStatus + wVZ + wCA + wAC + wNS + wPR + wRL
		+ wMW + wSize+ wTOF + wQoS + wAddr + wMPF1
		+ wRType + wRBID + wNdpAux + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wRBRMProt = ecc_width;
		}
		}

wRBUMProt	Integer	if (! ResilienceEnable) {wRBUMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wRBUMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wRBUMProt = 1;
		} else {
		auto temp = wCMStatus + wRL + wRBID + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) <)temp + ecc_width)) {
		ecc_width += 1;
		}
		wRBUMProt = ecc_width;
		}
		}

wDTRMProt	Integer	if (! ResilienceEnable) { wDTRMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wDTRMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wDTRMProt = 1;
		} else {
		auto temp = wCMStatus + wRL + wTM + wMPF1 + wNdpAux
		+ wRMessageId;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wDTRMProt = ecc_width;
		}
		}

Parameter Name	Type	Description/Derivation

wDTWMProt	Integer	if (! ResilienceEnable) {wDTWMProt = 0; }
		else { if (TIResiliencyProtectionType == NONE) {
		wDTWMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wDTWMProt = 1;
		} else {
		auto temp = wCMStatus + wRL + wTM + wPrimary + wMPF1 + wMPF2
		+ wRBID + wNdpAux + wIntfSize ;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wDTWMProt = ecc_width;
		}
		}

wDTWDBGMProt	Integer	if (! ResilienceEnable) {wDTWDBGMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wDTWDBGMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wDTWDBGMProt = 1;
		} else {
		auto temp = wCMStatus + wRL + wTM + wPrimary + wMPF1+ wMPF
		+ wRBID + wNdpAux+ wIntfSize;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wDTWDBGMProt = ecc_width;
		}
		}

25.4. System parameter : concertocresponsemessagefields

TABLE 25-4: CONCERTOCRESPONSEMESSAGEFIELDS PARAMETERS

Parameter Name	Type	Description/Derivation
wCCMDrspNdp	Integer	wCCMDrspNdp = wCMStatus + wRMessageId + wTM + wCCMDrspMProt
wSYSrspNdp	Integer	wSYSrspNdp = wCMStatus + wRMessageId + wTM + wSYSrspMProt
wNCCMDrspNdp	Integer	wNCCMDrspNdp = wCMStatus+ wRMessageId + wTM+ wNCCMDrspMProt
wSNPrspNdp	Integer	wSNPrspNdp = wCMStatus + wIntfSize + wMPF1 + wRMessageId + wTM + wSNPrspMProt
wDTWrspNdp	Integer	wDTWrspNdp = wCMStatus + wRMessageId + wRL + wTM + wDTWrspMProt
wDTWDBGrspNdp	Integer	wDTWDBGrspNdp = wCMStatus + wRMessageId + wRL + wTM + wDTWDBGrspMProt
wDTRrspNdp	Integer	wDTRrspNdp = wCMStatus + wRMessageId + wTM + wDTRrspMProt
wHNTrspNdp	Integer	
wMRDrspNdp	Integer	wMRDrspNdp = wCMStatus + wRMessageId + wTM + wMRDrspMProt
wSTRrspNdp	Integer	wSTRrspNdp = wCMStatus + wRMessageId + wTM + wSTRrspMProt;
wUPDrspNdp	Integer	wUPDrspNdp = wCMStatus+ wRMessageId+ wTM + wUPDrspMProt
wRBRrspNdp	Integer	wRBRrspNdp = wCMStatus + wRMessageId + wTM + wRBRrspMProt
wRBUrspNdp	Integer	wRBUrspNdp = wCMStatus + wRMessageId + wTM+ wRBUrspMProt

Parameter Name	Type	Description/Derivation
wCMPrspNdp	Integer	wCMPrspNdp = wCMStatus+ wRMessageId + wTM + wCMPrspMProt
wCMErspNdp	Integer	Not Used in Ncore3
wTUNrspNdp	Integer	Not Used in Ncore3
wTRErspNdp	Integer	Not Used in Ncore3

wCCMDrspMProt	Integer	if (! ResilienceEnable) { wCCMDrspMProt = 0; }
		else {
		if (TIResiliencyProtectionType == NONE) {
		wCCMDrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wCCMDrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wCCMDrspMProt = ecc_width;
		}
		}

wSYSrspMProt	Integer	if (! ResilienceEnable) {wSYSrspMProt = 0; }
		else {
		if (TIResiliencyProtectionType == NONE) {
		wSYSrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wSYSrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId +wTM
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wSYSrspMProt = ecc_width
		}
		}

wNCCMDrspMProt	Integer	if (! ResilienceEnable) {wNCCMDrspMProt = 0; }
		else {
		if (TIResiliencyProtectionType == NONE) {
		wNCCMDrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wNCCMDrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wNCCMDrspMProt = ecc_width;
		}
		}

Parameter Name	Type	Description/Derivation

wSNPrspMProt	Integer	if (! ResilienceEnable) {wSNPrspMProt = 0; }
		else {
		if (TIResiliencyProtectionType == NONE) {
		wSNPrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wSNPrspMProt = 1;
		} else {
		auto temp = wCMStatus + wIntfSize + wMPF1 + wRMessageId +wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;",
		}
		wSNPrspMProt = ecc_width;
		}
		}

wDTWrspMProt	Integer	if (! ResilienceEnable) {wDTWrspMProt = 0; }
		else {
		if (TIResiliencyProtectionType == NONE) {
		wDTWrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wDTWrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wRL + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;",
		};
		wDTWrspMProt = ecc_width;
		}
		}

wDTWDBGrspMProt	Integer	if (! ResilienceEnable) {wDTWDBGrspMProt = 0; }
		else { if (TIResiliencyProtectionType == NONE) {
		wDTWDBGrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wDTWDBGrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wRL + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < temp + ecc_width) {
		ecc_width += 1;
		}
		wDTWDBGrspMProt = ecc_width;
		}
		}

wDTRrspMProt	Integer	if (! ResilienceEnable) {wDTRrspMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wDTRrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wDTRrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wDTRrspMProt = ecc_width;
		}
		}

wHNTrspMProt	Integer	

Parameter Name	Type	Description/Derivation

wMRDrspMProt	Integer	if (! ResilienceEnable) {wMRDrspMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wMRDrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wMRDrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wMRDrspMProt = ecc_width;
		}
		}

wSTRrspMProt	Integer	if (! ResilienceEnable) {wSTRrspMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wSTRrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wSTRrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId +wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wSTRrspMProt = ecc_width;
		}
		}

wUPDrspMProt	Integer	if (! ResilienceEnable) {wUPDrspMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wUPDrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wUPDrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wUPDrspMProt = ecc_width;
		}
		}

wRBRrspMProt	Integer	if (! ResilienceEnable) {wRBRrspMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wRBRrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wRBRrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId +wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < (temp + ecc_width)) {
		ecc_width += 1;
		}
		wRBRrspMProt = ecc_width
		}
		}

Parameter Name	Type	Description/Derivation

wRBUrspMProt	Integer	if (! ResilienceEnable) {wRBUrspMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE) {
		wRBUrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY) {
		wRBUrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId +wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < temp + ecc_width) {
		ecc_width += 1;
		}
		wRBUrspMProt = ecc_width;
		}
		}

wCMPrspMProt	Integer	if (! ResilienceEnable) {wCMPrspMProt = 0; }
		else {if (TIResiliencyProtectionType == NONE)
		{wCMPrspMProt = 0;
		} else if (TIResiliencyProtectionType == PARITY)
		{wCMPrspMProt = 1;
		} else {
		auto temp = wCMStatus + wRMessageId + wTM;
		int64_t ecc_width = 3;
		while (std::pow(2, ecc_width - 1) < temp + ecc_width) {
		ecc_width += 1;
		}
		wCMPrspMProt = ecc_width;
		}
		}

26. *Legato Derived/Fixed Parameters*

26.1. PMA

A Power Management Adapter (PMA) will be instantiated, when power domains support dynamic control through a P-channel.

- If power domain is configured as dynamic (can be turned off by user), then a PMA will be allocated for all the clock domains inside of the power domain.
- If a power domain is configured as always on (will not be turned off in any case), then a PMA will be allocate when the clock domain can be turned off by a user signal (clock: external)
- PMA components do not have a CSR interface.

NOTE: PMA doesn't have any user settable/derived parameter for NCore 3.6.

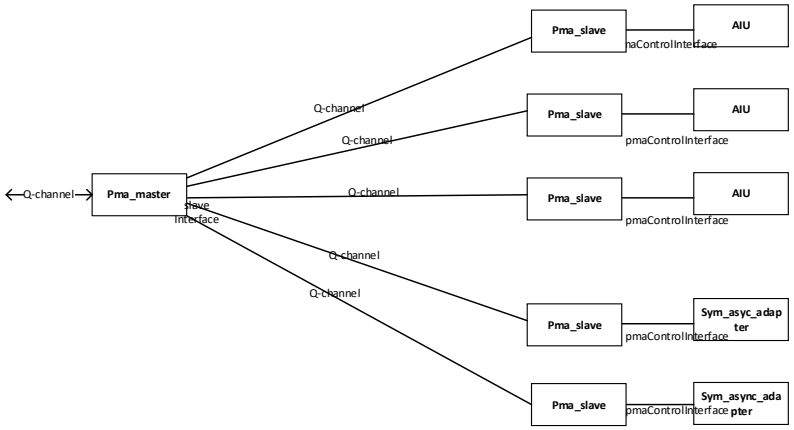


FIGURE 26-1: PMA IN CLOCK DOMAIN

26.2. Sym_async_adapter

TABLE 26-1: SYM_ASYNC_ADAPTER

Parameter Name	Default	Ranges	NCore 3.8	Comments
Async	false	True/False	Derived	
Depth			Derived	Circular FIFO depth of the sym_async_adpater would be derived by this system configuration value • syncDepth: 2 --> circular fifo depth of sym_async_adapter: 8 • syncDepth: 3 --> circular fifo depth of sym_async_adapter: 10 • syncDepth: 4 --> circular fifo depth of sym_async_adapter: 12
interfaces. inPmaControlInterface			Fixed	If IN clock interface is switchable this interface should exist. Otherwise, _SKIP_ = true.
interfaces. outPmaControlInterface			Fixed	If OUT clock interface is switchable this interface should exist. Otherwise, _SKIP_ = true.
interfaces. inProtectionInterface	_SKIP_ =True		Fixed	
Interfaces. outProtectionInterface	_SKIP_ =True		Fixed	

Depth:

- Depth parameter will be initially defined by Network parameter, and user will have override option.

Async:

- When the two clocks into sym_async adapter are from different clock domains, then async is set to true.
- When they are from different clock sub domains and the same clock domain, then async is set to false.

User will not be allowed to override this parameter.

26.3. Sym_buf_switch

All parameters for this element are not GUI visible.

TABLE 26-2: SYM_BUF_SWITCH

Parameter Name	Default	Ranges	NCore 3.6	Comments
arbType. egress	arb_rr1	arb_rr1, arb_pri_rr1, arb_fifo	Fixed	
bufLayer0. circular	*false*	True/False	**Derived**	Circular will be true when depth of the buffer is greater than 2.
bufLayer0. pipeForward	True	True/False	Fixed	If bufLayer1 and bufLayer2 have 0 depth, buflayer0 pipeForward must be true. (from CPR)
bufLayer2. circular	False	True/False	Fixed	
bufLayer2. depth	0	Power of two: Min:0 Max:32	Fixed	
bufLayer2. pipeBackward	True	True/False	Fixed	This will be fixed at NCore 3.6 but description added to let the readers know the default value
bufLayer2. pipeForward	True	True	Fixed	This will be fixed at NCore 3.6 but description added to let the readers know the default value
interfaces. protectionInterface			_SKIP_ = True (at R1)	
numPri	1		derived	

Circular parameter derivation:

Circular will be true when depth of the buffer is greater than 2

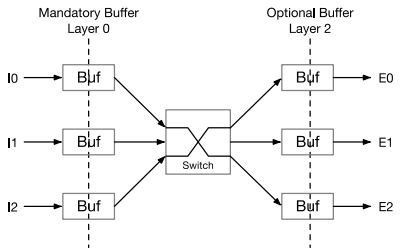


FIGURE 26-2: SYM_BUF_SWITCH IN CDTI, WITH ONLY ONE VC

26.3.1. Configuration details

Default configuration:

- bufLayer0.pipeForward = True
- bufLayer0.depth = 2
- bufLayer2.pipebackward = True
- bufLayer2.pipeForward =True
- bufLayer2.depth = 0

Circular parameter:

- Circular default value from Network: False

Circular = true/false does not affect function or performance, but timing and power. When circular is false, the output stage of the FIFO is always the same register, so it has better output timing. However, it has worse power, because when the FIFO is READ, all the registers with data get clocked as the data shifts forward. When circular is true, the FIFO uses read and write pointers, so only one register is being written or read at a time. It can have better power, because only the pointers and one register at most would clock in one cycle, but the output timing is worse, because there is a mux selecting which register to read for the output of the FIFO.

26.4. Sym_ibuf_switch

NOTE: sym_ibuf_switch is required to support mesh topology.

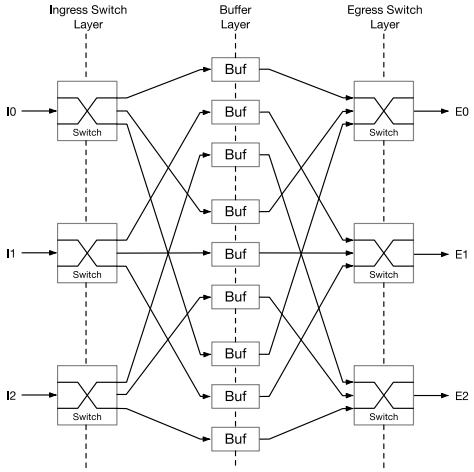


Figure 26-3: Sym_Buf_Switch in CDTI

TABLE 26-3: SYM_IBUF_SWITCH

Parameter Name	Default	Ranges	Ncore 3.8	GUI-Visibility	Comments
arbType.egress	arb_rr1	N/A	Fixed	No	arb_pri_rr1, arb_fifo would be valid values but are not supported currently
Circular	False	True/False	Derived	No	Circular will be true when depth of the buffer is greater than 2.
numPri	1		Derived	No	

26.5. Width/Rate_adapter (sym_nRate_adapter)

WidthAdapters

Ncore 3.x architecture supports different widths of networks between agents (64, 128, 256 bits) Connections between receivers and transmitters with different widths require a WidthAdapter.

A WidthAdapter converts a sequence of phits belonging to a packet arriving from a narrow interface to the wide interface.

This avoids using only part of the wide output interface's bandwidth, which would propagate downstream. A WidthAdapter will assemble a wider phit by storing:

- at least one phit entry of the width of the outgoing port
- one entry with the difference in width between the input and the output port

A WidthAdapter will introduce additional bubbles into the downstream traffic.
A WidthAdapter converting from wide interface to narrow interface may use a single wide entry to hold a phit while breaking it down into a stream of consecutive, narrow phits.

A WidthAdapter shall track up to 4 transactions and detect the boundary between packets having a different TxnID

TABLE 26-4: NINPUTWIDTH PARAMETERS FOR WIDTHADAPTER

Parameter Name	nInputWidth			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		Min: 64 Max: 512	Min: 64 Max: 512	
Constraint/Dependency	nInputWidth != nOutputWidth			
Customer Description				
Engineering Description	This value is a derived parameter based on the width of the source feeding this block. Maestro derives this parameter from the {FUnit, Switch}.sender.width connected to the input of the WidthAdapter.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 26-5: NOUTPUTWIDTH PARAMETERS FOR WIDTHADAPTER

Parameter Name	nOutputWidth			
Value	**Data Type**	**Architecture**	**Release**	**Default**
		Min: 64 Max: 512	Min: 64 Max: 512	
Constraint/Dependency	nInputWidth != nOutputWidth			
Customer Description				
Engineering Description	This value is a derived parameter based on the width of the source feeding this block. Maestro derives this parameter from the {FUnit, Switch}.receiver.width connected to the input of the WidthAdapter.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

Table 26-6: boolPipeline parameters for WidthAdapter

Parameter Name	*boolPipeline*			
Value	***Data Type***	***Architecture***	***Release***	***Default***
		True, False	True, False	*False*
Constraint/Dependency	nDepth ≤ 1			
Customer Description	Force insertion of at least one pipeline stage for timing reasons			
Engineering Description	Setting this parameter to True will override nDepth == 0 and force the insertion of at least one pipeline stage into the WidthAdapter. The setting has no effect if nDepth > 0.			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 26-7: NDEPTH PARAMETERS FOR WIDTHADAPTER

Parameter Name	*nDepth*			
Value	***Data Type***	***Architecture***	***Release***	***Default***
		Min: 0 Max: 4 x nTxnSize / nOutputWidth	Min: 0 Max: 4 x nTxnSize / nOutputWidth	
Constraint/Dependency	nDepth ≤ 1			
Customer Description	Add additional buffer stages to the WidthAdapter - this makes it a combined Width-Rate-Adapter			
Engineering Description	Add additional buffer stages to the WidthAdpater so that backpressure into the adapter will not immediately stall upstream, bubbles created by the width conversion will be squashed. The supported max. amount of buffer inserted will be 4 full transactions Note: TxnSize = 64 bytes = 512 bits			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

RateAdapters

Rate adapters will explicitly be instantiated by the user.

A RateAdapter will be used when a packet, consiting of multiple phits, may contain bubbles.
The rate adapter's function is, to aggregate temporally separated pieces/phits of a transaction, and retransmit them as consecutive sequence to a downstream receiver.
The Legato interconnect does not support transmission of flits belonging to different transactions.

Rate adapters may be used to level out fluctuations in input rate, even when the arrival rate ≥ departure rate for a short time, at the cost of increased buffering

- Rate adapters always have the same width on the input and the output port
- A rate adapter implements a FiFo-Queue where the first phit of a packet (flit) will not signal valid to the downstream receiver until the entire packet has been assembled in the queue.
- A rate adapter has to implement sufficient storage to hold at least one full packet - n buffer entries organized as width bits
- number of entries n = txn_size/port_width

Pipeline support shall be supported (improved timing), adding one additional storage entry of width bits to receive the first phit for the next transaction.
Additional entries may be specified if the designer desires to optimize bursty traffic in front of a congested switch.
This will support more than a single transaction to be forwarded in an uninterrupted burst.
A rate adapter will introduce additional latency of m cycles:

- m ≥ number of phits per transaction + 1

A width adapter shall track up to 4 transactions and detect the boundary between packets having a different TxnID.

TABLE 26-8: NWIDTH PARAMETERS FOR RATEADAPTER

Parameter Name	***nWidth***			
Value	***Data Type***	***Architecture***	***Release***	***Default***
		Min: 64 Max: 512	Min: 64 Max: 512	
Constraint/Dependency	nWidth == nInputWidth == nOutputWidth			
Customer Description				
Engineering Description	The width value is a derived parameter based on the width of both, the source feeding this block and the destination of the output. Maestro derives this parameter from the {FUnit, Switch}.sender.width connected to the input of the RateAdapter			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	Engineering	
Change History				

Field Code Changed

TABLE 26-9: NDEPTH PARAMETERS FOR WIDTHADAPTER

Parameter Name	***nDepth***			
Value	***Data Type***	***Architecture***	***Release***	***Default***
		Min: 0 Max: 4 x nTxnSize / nOutputWidth	Min: 0 Max: 4 x nTxnSize / nOutputWidth	
Constraint/Dependency	nDepth ≤ 1			
Customer Description	Defines the depth of the RateAdapter			
Engineering Description	Add additional buffer stages to the connection so that backpressure will not immediately stall upstream, bubbles in the stream will be squashed. The supported max. amount of buffer inserted will be 4 full transactions, the min. amount of buffer space will be 1 full transaction Note: TxnSize = 64 bytes = 512 bits			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

Software (Maestro) Support

Maestro shall support automated insertion of width adapters:
When source and destination of a network segment have different width
The decision shall be made based on:

- nInputWidth = {switch, FUnit} transmitter.width
- nOutputWidth = {switch, FUnit} receiver.width

The automatically generated WidthAdapter shall be customer configurable by changing the default settings of the following parameters:

- nDepth (default = 0) to configure additional buffer stages
- boolPipelined (default = false)

Maestro shall support user configurable insertion and removal of RateAdapters

- UI shall provide a means to select a network connection between two FUnits or an FUnit and a switch

The manually inserted RateAdapter shall be configurable by UI

- nDepth (default = TxnSize) to configure buffer stages
- nDepth shall be derived from the network segment where the user chose to insert the adapter
- When a user attempts to insert a rate adapter on a segment connecting a WidthAdapter output to a receiver, Maestro shall offer to parametrize the widthAdapter to increase depth instead (do we need a forced override to insert a RateAdapter?)
- When a user attempts to insert a rate adapter in front of a WidthAdapter - Maestro shall issue a warning, this is useless and only adds latency, recommend to parametrize the width adapter instead

Future versions of the RateAdapter may support different different clock domains for input and output ports.

TABLE 26-10: INSERTION RULES

	Input < Output	Input = Output	Input > Output	Description
Type	Width Adapter	Rate Adapter	Width/Rate Adapter	Adapter type depends on the interface configuration
Rule	*Automatic Insertion*	Insertion by User	Automatic Insertion	When input and output do not have the same width, a Width Adapter will be required
Configurability	Automatic Insertion = Yes nInputWidth nOutputWidth	Automatic Insertion = No	Automatic Insertion = Yes nInputWidth nOutputWidth	
	User boolPipeline nDepth[1]	User Insertion nWidth nDepth boolPipeline	User boolPipeline nDepth[1]	
nDepth	Automatic 1xnInputWidth + 1xnOutputWidth User +nxnOutputWidth	User Parameter based on rate difference ≥nxnWidth	Automatic ≥1xnInputWidth User +nxnOutputWidth	Automatic insertion will always use the minimum size required for the functionality User may configure additional storage in Maestro's UI
Notes: 1. Optional, additional buffer stages for rate adaptation				

26.6. Sym_pipe_adpater

NOTE: chapter 16.6

TABLE 26-11: SYM_PIPE_ADAPTER

Parameter Name	Default	Ranges	NCore 3.2	GUI-Visibility	Comments
Circular	true	True/False	Fixed	No	
Depth	2	**Power of two:** **Min:** - 1 **Max:** - 2	Fixed	Yes	
Split	false	True/False	Fixed	No	
interfaces.protectionInterface			_SKIP_ =True (at R1)	No	

26.7. Interrupt

Interrupts will not be aggregated within Ncore - the user needs to wire them outside of Ncore.

26.8. Parameter for CSR network

In the CSR network only atui_apb, atut_apb, and sym_switch/sym_buf_switch/ will be used.

After timing analysis, the user must insert sym_pipe_adapter manually.

No parameter will be visible to user.

No parameter is user settable. The next chapter is only for referring fixed values.

26.9. chi_async_adapter

sym_async_adapter is for SMI interface, and chi_async_adapter is to support CHI interface. It has a slave CHI interface and a master CHI interface, each interface has its own clock.

The circular FIFO depth of the chi_async_adpater is calculated according to the number of Chi request credit. (reqcredits = nCHIReqInFlight + 1.)

26.10. cxs_async_adapter

sym_async_adapter is for SMI interface, and cxs_async_adapter is to support CXS interface. It has a slave CXS interface and a master CXS interface, each interface has its own clock.

The circular FIFO depth of the cxs_async_adpater is calculated according to the number of cxs flit credit. (reqcredits = CXS_MAX_CREDIT + 1.)

26.11. CSR fixed parameters

This chapter describes fixed parameters for CSR network

26.11.1. Atut_apb parameters

TABLE 26-12: ATUT_APB BLOCK PARAMETERS

Parameter	Default	Ranges	CSR network	Description
apbSlvLut			Derived	
apbSlvLut.addr	*default*		Derived	
apbSlvLut.chipSel			Derived	
canReceiveNarrows	true	True/False	Derived	
ctlPipeCtxt	0	0 .. 9	0	
ctlPipeReq	0	0,1,2	0	
ctlPipeResp	0	0,1,2	0	
enBufWrite	false	True/False	False	
enPathLookup	false		Fixed true	When there is a tree structure in the CSR network, no route field is needed in the packet. Whether this is needed or not will be a function of the CSR network topology (would be required for a mesh depending on the routes used, even if there was only one initiator.)
exclusivesSupported	false		False	
fixedSupported	false		Fixed false	
idCompMask	[' true']		[] It must be fixed to an empty entry.	Not really applicable because APB doesn't have ID.
incrSupported	false		Fixed true	
Interfaces				
interfaces.apbInterface				
interfaces.apbRegInterface			No APB register interface	
interfaces.atpReqInterface			Derived	
interfaces.atpRespInterface			Derived	
interfaces.clkInterface			Derived	
interfaces.intInterface			Derived	
interfaces.pmaControlInterface			Derived	
interfaces.statsInterface			Derived	
mapBaseAddr	user		Derived	
mapBaseMask	user		Derived	
maxOutRd	1		Fixed as 1	

Parameter	Default	Ranges	CSR network	Description
maxOutTotal	1		Fixed as 1	
maxOutWr	1		Fixed as 1	
nExclEntries	4	2^N with N = 0 .. 3	Fixed 0	0 means no exclusive monitor.
narrowSupported	false		Fixed false	
nodeId	0		Derived	
numPri	1		Derived	
pathLut			Derived	
pathLut.route			Derived	
pathLut.targ_id			Derived	
pipeLevelApb	0	0,1,2	Fixed 2	For timing reasons this should be 1 or 2. This is a block level interface
pipeLevelAtp	0	0,1,2	Fixed 2	For timing reasons this should be 1 or 2. This is a block level interface
pipeLevelLut	0	0,1,2	Fixed 2	If a pathLut existed, should be 1 or 2.
pipeLevelSmi	2	0,1,2	Fixed 0	This could be 0. Internal interface
readInterleaveSupported	true		Fixed False	
rdEn	true	True/False	Always true	
smiDpknumPri	1		Derived	This needs to be the numPri for the CSR network , which should be 1
smiPktnumPri	1		Derived	This needs to be the numPri for the CSR network , which should be 1
timeoutErrChk	false		False	
timeoutErrCount	512		0	
timeoutUseExternalValue	0		Fixed 1	
wApbSlvDec	2		Derived	
wDataMax	64		Derived	This should be 32 bits. These are ignored when widthAdapterSupported = false
wDataMin	64		Derived	This should be 32 bits. These are ignored when widthAdapterSupported = false
wrEn	true	True/False	Always true	
widthAdaptionSupported	false		Derived	
wrapSupported	false		FALSE	

1533

26.11.2. Atui_axi parameters

1536 CSR column describes the value if the CSR would need different value from default parameter

TABLE 26-13: ATUI_AXI BLOCK PARAMETERS

Parameter	Default	Ranges	CSR Network	Description
axiPipeAr	2	0,1,2		User settable
axiPipeAw	2	0,1,2		User settable
axiPipeB	2	0,1,2		User settable
axiPipeR	2	0,1,2		User settable
axiPipeW	2	0,1,2		User settable
beatBufferEntries	0	5.2.2		User settable
ctlPipeCtxt	0	0 .. 9		User settable
ctlPipeReq	2	0,1,2		User settable
ctlPipeResp	2	0,1,2		User settable
enDecodeError	False		True	Fixed as True
enPathLookup	False		False	ATUI: fixed as false
enSplitting	False		False	Always True
idCompMask	[False]			Just use bottom bits. Fixed.
maxOutRd	8		2	User settable
maxOutTotal	2		2	User settable
maxOutWr	8		2	User settable
pipeLevel	2	0,1,2	0	User settable
pipeLevelAtp	2	0,1,2	2	User settable
pipeLevelLut	2	0,1,2	0	User settable
pipeLevelPam	0	0 to log2(maxPAMEntries)	2	User settable
pipeLevelRob	2	0,1,2	0	User settable
pmonStatsEn	False	True/False	False	User settable
rateLmtBktGlobal	8			User settable
rateLmtBktQueue_p	[0]			Fixed
rateLmtBktQueue_s	[0]			Fixed
rateLmtEn	False	True/False	False	User settable
rateLmtRefCntGlobal	8			User settable
rateLmtUseExternalValues	False			Fixed 1
refreshAmtGlobal	8			User settable
refreshAmtQueue_p	[0]			Fixed
refreshAmtQueue_s	[0]			Fixed
reorderingEntries	2		0	User settable
strpFunc	[' 0']		1	At R1, only struFunc = 1 will be used. Derived
timeoutErrCount	0	5.1.2	0	User settable
wRateLmtBktGlobal	0			Fixed
wRateLmtBktQueue	16			Fixed
wRateLmtRefCntGlobal	0			Fixed

Parameter	Default	Ranges	CSR Network	Description
wRateLmtRefCntQueue	16			Fixed
wRefreshAmtGlobal	0			Fixed
wRefreshAmtQueue	16			Fixed

26.11.3. APB socket parameters

TABLE 26-14: APB SOCKET PARAMETERS

Parameter Name	Default	Range	CSR Network	Description/Comment
wData	32	8, 16, 32, 64	32	
wAddr	12	minimum: 12 maximum: 64	12	This can the packet field width can be 12, because once a packet is headed toward a block on the CSR network, only the bottom 12 bits are needed. Bits above 12 are needed to select the block.
wPSel	1	1, 2, 4, 8, 16	1	
wStrb	0	APB2: 0 APB3: wData/8	wData/8	
wPSlverr	0	APB2: 0 APB3: 0 or 1	1	
wProt	0	APB2: 0 APB3: 3	3	
csrAccessSupported	True	True/False	False	
readSupported	True	Fixed true	True	
writeSupported	True	Fixed true	True	

26.11.4. AXI socket parameters

1545

TABLE 26-15: AXI SOCKET PARAMETERS

Parameter Name	Default	Range	CSR network	Description/Comment
wAddr	32	Minimum: 12 Maximum: 64	24	
wArUser	0	Minimum:0 Maximum: 64	0	
wArID	1	Minimum:1 Maximum:32	0	
wAwUser	0	Minimum:0 Maximum: 64	0	
wAwId	1	Minimum:1 Maximum:32	0	
wWUser	0	(wData/8 * 0, 1, 2, 3, 4, and 5) wWUser should be provided not the above but it is actual width. For example, if the data width is 64, and the user bit per byte is 1, wWUser should be 8. if the writeSuported = 0, wWuser = 0	0	
wData	32	8, 16, 32, 64, 128, 256, 512, 1K, 2K	32	
wRuser	0	(wData/8 * 0, 1, 2, 3, 4, and 5) wRUser should be provided not the above but it is actual width. For example, if the data width is 64, and the user bit per byte is 1, wRUser should be 8. if the readSuported = 0, wRuser = 0	0	
wBuser	0	Minimum:0 Maximum: 64	0	
wLen	8	AXI3: 4 [3:0] AXI4: 8 [7:0]	4	
wSize	3	Minimum:3 Maximum:4	3	
wLock	1	AXI3: 2 [1:0] AXI4: 1	1	
wProt	3	3 [2:0]	3	
wQos	4	AXI3: 0 AXI4: 4	0	

Parameter Name	Default	Range	CSR network	Description/Comment
wRegion	0	It is only in AXI4: Minimum:0 Maximum:4	0	
nativeType	Axi4	Axi3/Axi4	Axi4	
eAr	1	0,1	1	
eAw	1	0,1	1	
csrAccessSupported	true	True/False	False	
wrapSupported	false	True/False	False	
narrowSupported	false	True/False	False	
fixedSupported	false	True/False	False	
readSupported	True	True/False	True	
writeSupported	True	True/False	True	
readInterleaveSupported	False	True/False	False	
earlyWriteReponseSupported	False	True/False	False	
maxBurstLength	16	2^N with N = {0 .. 12}	1	

26.11.5. Switch parameters

Network parameters will be fixed. Also, block parameters for all the switches will be fixed

- Only packet parallel style supported at NCore 3.2.
- Only sym_buf_switch will be used.

TABLE 26-16: SWITCH BLOCK PARAMETERS

Parameter		CSR	
defaultArbPolicy	sym_buf_switch	arb_rr1	
defaultInputSwitchDepth	sym_buf_switch	2	BufLayer0.depth
defaultOutputSwitchDepth	sym_buf_switch	0	BufLayer2.depth

27. *Other User Settable Parameters*

27.1. Parameter related with Placeholder Generic Signal

TABLE 27-1: PARAMETER FOR GENERIC PORT: WIRENAME

Parameter Name	wireName			
Value	Data Type	Architecture	Release	Default
Constraint/Dependency				
Customer Description	portName for Generic port			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 27-2: PARAMETER FOR GENERIC PORT: WIREWIDTH

Parameter Name	wireWidth			
Value	Data Type	Architecture	Release	Default
Constraint/Dependency				
Customer Description	Port width for generic port			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 27-3: PARAMETER FOR GENERIC PORT: WIRERTLPREFIX

Parameter Name	wireRtlPrefix			
Value	Data Type	Architecture	Release	Default
Constraint/Dependency				
Customer Description	RTL Prefix. For a given block, all the ports must have the same witeRtlPrefix.			
Engineering Description				
Release Info	Status	Effective version	Visibility	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 27-4: PARAMETER FOR GENERIC PORT: DIRECTION

Parameter Name	direction			
Value	Data Type	Architecture	Release	Default
	Valid Values	[In, Out]		In

Constraint/Dependency			
Customer Description	Parameter for port direction configuration		
Engineering Description			
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	User-GUI
Change History			

Field Code Changed

27.2. Parameter related with SRAM assignment

27.2.1. SW_memory

TABLE 27-5: MEMORY PARAMETER FOR IOAIU

Memory Name	Constraints	Number of Memories
OttMem	*MemoryProtectionType (**Error! Reference source not found.**) cannot be "NONE"* *memoryType must be SRAM*	Same as #.of nOttDataBanks (Chapter 16.1)

TABLE 27-6: MEMORY PARAMETER FOR SNOOP FILTER

Memory Name	Constraints	Number of Memories
TagMem	MemoryProtectionType (**Error! Reference source not found.**) cannot be "NONE" memoryType must be SRAM	Same as #.of nWays (Chapter 11.3) -- bitEn == 0 in NCore 3.2.

TABLE 27-7: MEMORY PARAMETER FOR DCE

Memory Name	Constraints	Number of Memories
skidBufferMem	Default value is FLOP, and it can be set as SRAM. If it is set as SRAM, Maestro needs to pass this memory object with 1R1W type SRAM to DCE. MemoryProtectionType (**Error! Reference source not found.**) cannot be "NONE", memoryType must be SRAM	1

TABLE 27-8: MEMORY PARAMETER FOR DMI

Memory Name	Constraints	Number of Memories
writeDataMem	MemoryProtectionType (**Error! Reference source not found.**) cannot be "NONE" memoryType must be SRAM	Same as # of nCohWrDataBanks (Chapter 13.1)
rdDataMem	MemoryProtectionType (**Error! Reference source not found.**) cannot be "NONE" memoryType must be SRAM	2
CMDReqSbMem	Default value is FLOP, and it can be set as SRAM. If it is set as SRAM, Maestro needs to pass this memory object with 1R1W type SRAM to DMI. MemoryProtectionType (**Error! Reference source not found.**) cannot be "NONE", memoryType must be SRAM	1
MRDReqSbMem	Default value is FLOP, and it can be set as SRAM. If it is set as SRAM, Maestro needs to pass this memory object with 1R1W type SRAM to DMI. MemoryProtectionType (**Error! Reference source not found.**) cannot be "NONE", memoryType must be SRAM	1

TABLE 27-9: MEMORY PARAMETER FOR CCP

Memory Name	Constraints	Number of Memories
TagMem	MemoryProtectionType (**Error! Reference source not found.**) cannot be "NONE" memoryType must be SRAM	Same as #.of nTagBanks
DataMem	MemoryProtectionType (**Error! Reference source not found.Error! Reference source not found.**) cannot be "NONE" memoryType must be SRAM	Same as #.of nDataBanks

TABLE 27-10: MEMORY PARAMETER FOR DVE

Memory Name	Constraints	Number of Memories
TraceMem	MemoryProtectionType (**Error! Reference source not found.**) cannot be "NONE" MemoryType must be SRAM	2

TABLE 27-11: MEMORY PARAMETER FOR DII

Memory Name	Constraints	Number of Memories
skidBufferMem	Default value is FLOP, and it can be set as SRAM. If it is set as SRAM, Maestro needs to pass this memory object with 1R1W type SRAM to DII. MemoryProtectionType (**Error! Reference source not found.**) cannot be "NONE", memoryType must be SRAM	1

TABLE 27-12: ENHALFSPEEDDATASRAM PARAMETER

Parameter Name	enHalfSpeedDataSRAM			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 1	Min: 0 Max: 1	0
Constraint/Dependency	Only available in DMI with SMC enabled			
Customer Description	Enable SMC data SRAM to run at half clock frequency. Enabling this will add a couple of cycle latency and may affect BW in the case of partial cache line accesses.			
Engineering Description	This applies to only DMI with SMC enabled			
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.6	User-GUI	
Change History				

TABLE 27-13: ENSRAMPIPE PARAMETER

Parameter Name	enSRAMPipe			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 0 Max: 1	Min: 0 Max: 1	0
Constraint/Dependency	Available in DMI with SMC enabled Available in IOAIU for OTT data SRAM In the case of DMI this must be enabled if enHalfSpeedDataSRAM is set.			
Customer Description	Enable SRAM pipe. Enabling this will add a cycle latency.			

Field Code Changed

Engineering Description	Enable SRAM pipe. Enabling this will add a cycle latency. In the case of DMI this must be enabled if enHalfSpeedDataSRAM is set.		
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	User-GUI
Change History			

Field Code Changed

27.2.2. Generic ports

Generic ports are used to create user defined signals for SRAM interfaces. This is a common for all blocks.

Software supports definition of N generic ports of width m ≤ 1023 bits for each block that instantiates memories - need to check if this is implemented, how is it verified, ports created and verified connected all the way through the hierarchy - port reaches all the way to the top level, DFT chimney for DFT signals - user instantiates memory wrapper with his DFT logic - these signals will be used to bring these signals up - need to use the same name as the ports on the memory wrapper etc.

TABLE 27-14: PARAMETER FOR SRAM GENERIC PORT: WIRENAME

Parameter Name	wireName			
Value	**Data Type**	**Architecture**	**Release**	**Default**
Constraint/Dependency				
Customer Description	portName for Generic port			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 27-15: PARAMETER FOR SRAM GENERIC PORT: WIREWIDTH

Parameter Name	wireWidth			
Value	**Data Type**	**Architecture**	**Release**	**Default**
			Max: 1024	
Constraint/Dependency				
Customer Description	Port width for generic port Maximum width per signal in generic interface is 1024			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 27-16: PARAMETER FOR SRAM GENERIC PORT: DIRECTION

Parameter Name	direction			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	[In, Out]		In

Constraint/Dependency			
Customer Description			
Engineering Description			
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	User-GUI
Change History			

Field Code Changed

28. *User Settable parameter for Synthesis*

TABLE 28-1: SYNTHESIS PARAMETER: CHECKONLY

Parameter Name	checkOnly			
Value	**Data Type**	**Architecture**	**Release**	**Default**
Constraint/Dependency				
Customer Description	Whether to stop the RTL flow after compilation and linking, or to proceed to synthesis			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 28-2: SYNTHESIS PARAMETER: TOPOMODE

Parameter Name	topoMode			
Value	**Data Type**	**Architecture**	**Release**	**Default**
Constraint/Dependency				
Customer Description	Whether to launch the synthesis tools in topographical mode, or WLM. The latter is faster but less accurate			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 28-3: SYNTHESIS PARAMETER: TECHNOLOGY

Parameter Name	technode			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Valid Values	ERROR,TSMC16,TSMC7, CUSTOM		CUSTOM
Constraint/Dependency				
Customer Description	Custom generates a technology template file which contains the variables which need to be filled in with the users technology library information before running synthesis			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 28-4: SYNTHESIS PARAMETER: CLOCKUNCERTAINTY

Parameter Name	clockUncertainty			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer	Min: 1 Max: 99		15
Constraint/Dependency				
Customer Description	The default clock uncertainty to assume for clocks, as a percentage (e.g. 15 = 15%). The value can be overwritten in the generated synthesis scripts.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 28-5: SYNTHESIS PARAMETER: RTLWRAPPERDIR

Parameter Name	rtlWrapperDir			
Value	**Data Type**	**Architecture**	**Release**	**Default**
Constraint/Dependency				
Customer Description	Directory with user-written Verilog files which instantiate custom cells, such as memories. They override generic-behavior Verilog files generated by Maestro (which implement memories as a \"sea of registers\") and must be named identically.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 28-6: SYNTHESIS PARAMETER: HARDMACRODBS

Parameter Name	hardMacroDbs			
Value	**Data Type**	**Architecture**	**Release**	**Default**
Constraint/Dependency				
Customer Description	Specifies the location and names of the hard macros in the design, such as compiled memories			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 28-7: SYNTHESIS PARAMETER: BOTTOMUPSYNTHESIS

Parameter Name	bottomUpSynthesis			
Value	**Data Type**	**Architecture**	**Release**	**Default**
				True
Constraint/Dependency				
Customer Description	If selected will write out scripts and hierarchy for a bottom of synthesis run, with all the Ncore units written out to their own directories. If not selected a top down, flat synthesis hierarchy will be gnerated.			

Engineering Description			
Release Info	**Status**	**Effective version**	**Visibility**
	Active	3.2	User-GUI
Change History			

Field Code Changed

TABLE 28-8: SYNTHESIS PARAMETER: MAXTRANSITION

Parameter Name	maxTransition			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer			150
Constraint/Dependency				
Customer Description	Default transition delay on functional input ports (THIS SHOULD BE RENAMED)			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 28-9: SYNTHESIS PARAMETER: OUTPUT LOAD

Parameter Name	outputLoad			
Value	**Data Type**	**Architecture**	**Release**	**Default**
	Integer			100000
Constraint/Dependency				
Customer Description	The default capacitive load on functional output ports			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 28-10: SYNTHESIS PARAMETER: ULVTPERCENTAGE

Parameter Name	ulvtPercentage			
Value	**Data Type**	**Architecture**	**Release**	**Default**
Constraint/Dependency				
Customer Description	Ulvt Percentage: Ulvt percentage limit set in synthesis scripts.			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed

TABLE 28-11: SYNTHESIS PARAMETER: COMPILECOMMAND

Parameter Name	compileCommand			
Value	**Data Type**	**Architecture**	**Release**	**Default**
Constraint/Dependency				
Customer Description	Compile command: Allows the user to add in options to default synthesis command			
Engineering Description				
Release Info	**Status**	**Effective version**	**Visibility**	
	Active	3.2	User-GUI	
Change History				

Field Code Changed