Ncore 3.7 - Architecture Parameter Documentation

Rev: 1.08, September 26, 2025

ARTERIS® NCORE 3.7 - ARCHITECTURE PARAMETER DOCUMENTATION

Release Information

Version	Editor	Change	Date
1.08	HL	• Added Table 7-20 and Table 7-52 to enable Trace_Signals property always true once AXI5 and ACE5 interfaces are used per discussed in [CONC-17734]	9/26/2025
1.07	HL	• Updated the following per identified in [CONC-17599] ○ Sync-up CSR starting address as 0x2e80_0000 and updated it according in Table 6-1 ○ Updated the default memorySize in boot-region to be 16 KB in Table 6-4 ○ Updated nAttCtrlEntries of DCE is maximum at 96 in Table 10-1 ○ Updated nAiuPorts maximum at 8 in Table 4-3 ○ Added Table 11-20 for the missing useAtomic parameter ○ Updated nSets maximum size for snoop filter in Table 10-7 and corresponding descriptions also per requested in [NCORE-839] ○ Added Table 11-24 for the missing addressBits parameter for DMI	9/3/2025
1.06	HL	• Added Trace_Signals property to sync up with what Maestro and hardware can support for ACE5-Lite and ACE5-LiteDVM interfaces and leave AXI5 interface as is (Not supported) per discussed in [CONC-17255] ○ Added Table 7-71	6/6/2025
1.05	HL	• Clarified Cache_Stash_Transactions property usage in Table 7-70 per discussed in [CONC-17146] and [CONC-17155] • Changed IOAIUp configuration as PCAIU wherever applicable per described in [MAES-8134]	5/18/2025
1.03	HL	• Updated the minimum value of nOttCtrlEntries to be 8 in Table 9-1 as per identified in [CONC-16916] to align with the design intent • Updated and added AXI5 into Atomic supports in section 11.5 per discussed in [MAES-8107]	4/28/2025
1.02	HL	• Updated the minimum value of nOttCtrlEntries in Table 14-3 to be 8 as per identified in [CONC-16554] to align with the design intent.	4/10/2025
1.01	HL	• Changed LPID width from 8 to 5 to be compliant with IP-XACT as identified in [CONC-16302] in Table 18-8 • Also added a new field called GroupExtID with a width of 3 to keep the size of reqflit still as 8, which is defined as of {GroupExtID, LPID}	3/31/2025
1.00	HL	• Set to version 1.00 to sync up with product release	3/20/2025
0.87	HL	• Made SyscoReq/ACK interface always presents per described in [CONC-15595] for both CHI-B/CHI-E protocols by updating the following: ○ Table 7-87: parameter related to sysco interface of CHI-B ○ Table 7-96: parameter related to sysco interface of CHI-E	1/31/2025
0.86	HL	• Added more clarifications in Table 5-1 on frequency selection and its recommended range per described in [MAES-7558] • Updated Table 7-55 and Table 7-59 to add address width check when DVM_v8 or better is enabled per discussed in [CONC-14342]	1/24/2025
0.85	HL	• Added the missing global parameter “useRtlPrefix” in Table 4-31: useRtlPrefix parameter per identified in [CONC-15282] • Made Exclusive_Accesses property always supported by default and not user visible per discussed in [CONC-16300], the following tables are updated accordingly: ○ Table 7-8: Exclusive_Accesses Property for AXI4 Interface ○ Table 7-16: Exclusive_Accesses Property for AXI5 Interface ○ Table 7-27: Exclusive_Accesses Property for ACE Interface ○ Table 7-46: Exclusive_Accesses Property for ACE5 Interface ○ Table 7-68: Exclusive_Accesses Property for ACE5-Lite/ACE5-LiteDVM Interface ○ Table 7-81: Exclusive_Accesses Property for ACE-Lite Interface ○ Table 18-1: AXI Interface Fixed Parameters ○ Table 20-15: AXI Socket Parameters	1/9/2025

0.83	HL	• Made “shareableTransactions” not user visible and default as TRUE to enable a seamless transition adaption on this property per aligned in [MAES-7834]. The following tables are updated accordingly: ○ Table 7-30: Shareable_Transactions Property for ACE Interface ○ Table 7-47: Shareable_Transactions Property for ACE5 Interface ○ Table 7-69: Shareable_Transactions Property for ACE5-Lite/ACE5-LiteDVM Interface ○ Table 7-80: Shareable_Transactions Property for ACE-Lite Interface ○ Table 18-3: ACE Interface User Settable Parameters ○ Table 18-4: ACE5-LITE Interface Derived Parameters ○ Table 18-5: ACE-LITE Interface Derived Parameters • Made COHERENCY_CONNECTION_SIGNALS property not user visible and always FALSE to enable a seamless adaption on this property per aligned in [MAES-7849]. The following tables are updated accordingly: ○ Table 7-19: Coherency_Connection_Signals Property parameter for AXI5 interface ○ Table 7-35: Coherency_Connection_Signals Property parameter For ACE interface ○ Table 7-42: Coherency_Connection_Signals Property parameter for ACE5 interface ○ Table 7-59: Parameter to Enable DVM Functionalities for the ACE5-LITE Interface: DVM ○ Table 7-66: Coherency_Connection_Signals Property parameter For ACE5-LiteDVM interface	1/7/2025
0.82	HL	• Corrected and updated low and medium priority feedback per identified in [CONC-16120] ○ Made wSize parameter in Table 20-15: AXI Socket Parameters fixed at 3 ○ Further clarified the usage of “noDVM” parameter in section 4.9 ○ Corrected the constraint range to be “1, 2, 4” in Table 14-12 and Table 11-16 ○ Corrected CSR network components usage in section 20.8 ○ Other miscellaneous typos and wrong references across the change records and specification	12/30/2024
0.81	HL	• Corrected and updated the medium and high priority feedback per identified in [CONC-16120] ○ A wrong reference to the system architecture specification for the connectivity mapping in section 4.2 ○ A wrong reference to the system architecture specification for the Definition of a Memory Address Map to in section 4.3 ○ Updated properly for the sym_nRate_adapter support in section 17.4 ○ A wrong reference to the system architecture specification for Functional Safety in section 4.4 ○ Updated a few explicit references to Ncore 3.2/3.4/3.6 by “Starting with Ncore 3.2” in section 6	12/20/2024
0.80	HL	• Updated Table 14-11 to make maximum number of Tag Bank to be 2 per discussed in [CONC-15937]	12/9/2024
0.79	BH	• Update Table 4-4, Table 4-5 , change type to int [CONC-15872] • Update Table 4-30, modify the visibility to user [CONC-15876] • Update Table 5-1, modify clock frequency range [CONC-15873] • Update Table 5-3, modify the clock gating description [CONC-15875] • Update Table 7-97, Table 7-98, modify the naming typo [CONC-15877] • Update Table 10-12, Table 10-13, modify JSON typo [CONC-15880] • Update Table 15-8. Remove customer description as the parameter is not visible to user [CONC-15879]	11/25/2024
0.78	HL	• Updated Table 4-20 and added more descriptions to clarify the QoS value mapping to those used inside of an Ncore per discussed in [CONC-15437] • Per requested in [CONC-15824]	11/12/2024

		○ Removed previous Table 10-4, Table 11-7, Table 11-10 and Table 12-7 per requested into streamline the SRAM usage for Skid Buffer across DCE, DMI and DII; ○ Also, updated the definition and usage for the memory object defined in Table 21-7, Table 21-8 and Table 21-11 to take FLOP as the default value, and make SRAM as an option.	
0.77	HL	• Further cleanup and sync up on the Coherency_Connection_Signals property usage per ARM AMBA IP-XACT definitions as per discussed in [MAES-7632] ○ For ACE, ACE-Lite, AXI4 with ProxyCache configured, it is NOT a valid property, this property is set as FALSE always (not a user settable feature), and agent connected with NCAIU can only go through the handshake via the usage of CSR. Updated the following tables to make this parameter always False and not visible to user: ▪ Table 7-9: Coherency_Connection_Signals Property parameter For AXI4 interface ▪ Table 7-35: Coherency_Connection_Signals Property parameter For ACE interface ▪ Table 7-79: Coherency_Connection_Signals Property parameter for ACE-Lite interface ○ For AXI5, ACE5 and ACE5-LiteDVM, it is a valid property, this property is ser settable but default as FALSE. Thus, the handshake will be carried out via the usage of CSR. Updated the following tables ▪ Table 7-19: Coherency_Connection_Signals Property parameter for AXI5 interface ▪ Table 7-42: Coherency_Connection_Signals Property parameter for ACE5 interface ▪ Table 7-66: Coherency_Connection_Signals Property parameter For ACE5-LiteDVM interface ○ For IOAIUp configuration, this property is NOT applicable since it doesn't allow a ProxyCache to be configured together with the usage of OWO. • Relaxed the SRAM usage by removing the 256-entry minimum size requirement per described in [CONC-15796]	11/8/2024
0.76	HL	• Added the following properties for ACE5 interface in section 7.5 ○ Check_Type ○ Exclusive_Accesses ○ Coherency_Connection_Signals ○ DVM_v8 ○ DVM_v8.1 ○ DVM_Message_Support ○ Shareable_Transactions ○ Continuous_Cache_Line_Read_Data • Added the following properties for AXI5 interface in section 7.3 ○ Check_Type ○ Atomic_Transactions ○ Exclusive_Accesses ○ Coherency_Connection_Signals (when hasProxyCache is enabled) • After a review of Benard Bonardi, added the following properties for AXI4 interface in section 7.2 ○ Exclusive_Accesses ○ Coherency_Connection_Signals (when hasProxyCache is enabled) • Added the following properties for ACE interface in section 7.4 ○ Exclusive_Accesses ○ DVM_v8 ○ DVM_v8.1	10/17/2024

		○ DVM_Message_Support ○ Shareable_Transactions ○ *Coherency_Connection_Signals* ○ Continuous_Cache_Line_Read_Data • Changed the parameter name of eAC to DVM defined in Table 7-59 to make it as a more meaningful name from now on • Added the following properties for ACE5-Lite/ACE5-LiteDVM interface in session 7.6 ○ Atomic_Transactions ○ Check_Type ○ Exclusive_Accesses ○ Shareable_Transactions ○ The following properties only applicable once DVM is enabled to make the interface as an ACE5-LiteDVM ▪ Coherency_Connection_Signals ▪ DVM_v8 ▪ DVM_v8.1 ▪ DVM_v8.4 ▪ DVM_Message_Support	
0.75		• Updated Table 7-1 to add AXI5 and ACE5 as two native interfaces that will be supported beyond 3.7 based on the agreement made in a group discussion on September 27, 2024 • Took out check_type support in ACE, ACE-Lite and AXI as described in [CONC-15615] by removing the previous ○ Table 7-8 for AXI ○ Table 7-18 for ACE ○ Table 7-35 for ACE-Lite	10/11/2024
0.74	HL	• Added missing parameters related to proxy cache configuration with four tables from Table 14-10 to Table 14-14 per identified in [CONC-15529] and [NCOR-612] • Updated accordingly for the following tables related to nLatencyCounters" per described in [CONC-15513] ○ Added the reason why nPerfCounters value maximum is 8 for CAIU and NCAIU in section 9.5 and section 14.7 ○ Updated Table 10-14,Table 11-18,Table 12-9 and Table 13-3 to make the maximum performance counter value to be 16. • Changed useSysCoInt default value from TRUE to FALSE per discussed in [CONC-15503]	10/7/2024
0.73	HL	• Added memory object parameter for SRAM implementation of SkidBuffer in DII as defined in Table 21-11 • Added memory object parameter for SRAM implementation of SkidBuffer in DCE as defined in Table 21-7 per required in [MAES-7547] • Added memory object parameter for SRAM implementation of SkidBuffer as defined in Table 21-8 for DMI per required in [MAES-7560] • Updated Table 14-29 to make it as an internal parameter based on a discussion meeting on how to handle IOAIUp data width held on September 13, 2024	9/20/2024
0.72	HL	• Updated and clarified DII usage in Table 9-20 as described in [MAES-7432] • Updated and take out EventOutInt support for ACE-Lite since it is not supported per described in [CONC-6823]	9/13/2024
0.71	HL	Minor updates as the following based on the group review held on September 5, 2024: • Added more clarifications on sym_ibuf_switch in section 20.4. • Added definitions of SID and TID in Table 16-2.	9/5/2024
0.70	HL	• Updated Table 10-2, Table 11-5, Table 11-7 and Table 12-5 to increase maximum size of skid buffer from 320 to 768, which implies the maximum	9/4/2024

		size of the skid overflow buffer is increased from 256 to 512 per requested in [CONC-14930] and [CONC-15290]	
0.60	HL	• Removed watermark to improve better reading and reviewing experience. • Removed asyncInterface parameter defined in Table 14-28 after an internal review with Maestro team on August 30, 2024. At the same review meeting, it is agreed to replace asyncInterface parameter by the IOAIUpDataWidth parameter to enable the asynchronous gasket insertion and the data width adaption gasket insertion by Maestro. • Added sym_ibuf_switch support in section 20.4. • Added a new section 16, where all relevant parameters related to tiling are defined as requested in group review held on August 28, 2024 and suggestions from the designer.	8/30/2024
0.56	HL	• Added Table 4-10 and Table 4-15 to fix issue identified in [CONC-15245] • Added previous Table 12-7 to enable Skid Buffer SRAM implementation for a DII per requested by [CONC-15270] • Added Table 14-28 and Table 14-29 to add necessary parameters required to do the frequency and width adaption for OWO (PCIe acceleration).	8/16/2024
0.53	HL	• Updated the descriptions on previous Table 10-4, Table 11-7 and Table 11-10 Skid Buffer SRAM implementation to make it more consistent per group review feedback received on August 7, 2024 • Updated OWO feature limitations and constraints in Table 14-26 to reference system architecture specification for the details per group review feedback received on August 7, 2024 • Added more clarifications on the latency monitoring usage in a DCE in section 10.4.	8/12/2024
0.52	HL	• Updated and clarified the performance counter enabling and configuration in Table 9-11 and Table 9-12 for CAIU, Table 14-15 and Table 14-16 for NCAIU, Table 11-18 and Table 11-19 for DMI, Table 12-9 and Table 12-10 for DII as per described in [CONC-15156] • Added checkType parameter in Table 7-99 for CHI-E interface to enable Odd Parity check as per identified in [CONC-15202] • Added checkType parameter defined in Table 7-8 to enable the Odd Parity check for an AXI interface, added checkType parameter defined in Table 7-18 to enable the Odd Parity check for an ACE interface, added checkType parameter defined in Table 7-65 to enable the Odd Parity check for an ACE5-Lite interface and added checkType parameter defined in Table 7-35 to enable the Odd Parity check for an ACE-LITE interface per requested in [NCOR-605] • Added checkType parameter defined in Table 7-90 to enable Odd Parity check for a CHI-B interface per requested in [NCORE-543]	8/7/2024
0.51	HL	• Increased the maximum RbCredits from 32 to 64 for DCE in Table 10-4,Table 10-22 as per requested in [CONC-15038] • Increased the maximum value of nTagBank (the bank number of the SMC Tag memory) from 2 to 4 in Table 11-15 per requested in [CONC-14927] • Updated the constraints and descriptions on the SysCo and Event interfaces across all major interface protocols Ncore supports in Table 7-28,Table 7-29,Table 7-64,Table 7-89,Table 7-96,Table 7-97,Table 7-98 etc. per discussed in [MAES-6823] • Also updated in Table 7-78 that DVM functions are not applicable for ACE-Lite interface per discussed in [MAES-6823] • Added section 14.11 for OWO feature to accelerate PCIe traffic per requested in [CONC-549] and [NCORPM-71] • Added parameters described in previous Table 10-4, Table 11-7 and Table 11-10 to support the usage of SRAM for the Overflow FIFO of the Skid Buffer per requested in [NCOR-249][NCORPM-118]	7/25/2024
0.50	HL	This is the version that contains last few updates on version 3.6.4 and the base of version 3.7 • Updated Table 13-5 to reflect the max DVM credits can be set with the consideration of noDVM as per discussed in [MAES-7390]	7/17/2024
Legend:			

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

Note:

Issues to be discussed:

Confidential Proprietary Notice

This document is CONFIDENTIAL AND PROPRIETARY to Arteris, Inc. or its applicable subsidiary or affiliate (collectively or as applicable, “Arteris” or “Arteris IP”), and any use by you is subject to the terms of the agreement between you and Arteris IP or the terms of the agreement between you and the party authorized by Arteris IP to disclose this document to you.

This document is also protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arteris IP. **No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated. You are prohibited from altering or deleting this notice from any use by you of this document.**

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information: (i) for the purposes of determining whether implementations infringe any third party patents;(ii) for developing technology or products which avoid any of Arteris IP's intellectual property; or (iii) as a reference for modifying existing patents or patent applications or creating any continuation, continuation in part, or extension of existing patents or patent applications; or (iv) for generating data for publication or disclosure to third parties, which compares the performance or functionality of the Arteris IP technology described in this document with any other products created by you or a third party, without obtaining Arteris IP's prior written consent.

THIS DOCUMENT IS PROVIDED “AS IS”. ARTERIS IP PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arteris IP makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, third party patents, copyrights, trade secrets, or other rights. This document may include technical inaccuracies or typographical errors. Arteris IP makes no representations or warranties against the risk or presence of same.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARTERIS IP BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARTERIS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be solely responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arteris IP's customers is not intended to create or refer to any partnership relationship with any other company. Arteris IP may make changes to this document at any time and without notice. If any of the provisions contained in these terms conflict with any of the provisions of any click through or signed written agreement covering this document with Arteris IP, then the click-through or signed written agreement prevails over and supersedes the conflicting provisions of these terms. This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the agreement shall prevail.

The Arteris IP name and corporate logo, and words marked with ® or ™ are registered trademarks or trademarks of Arteris (or its subsidiaries) in the US and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arteris IP's trademark usage guidelines, available from Arteris IP upon request by emailing to contracts@arteris.com.

Confidentiality Status

Product Status

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

Web Address

http://www.arteris.com

Table of Contents

Table of Figures

1. *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.

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.

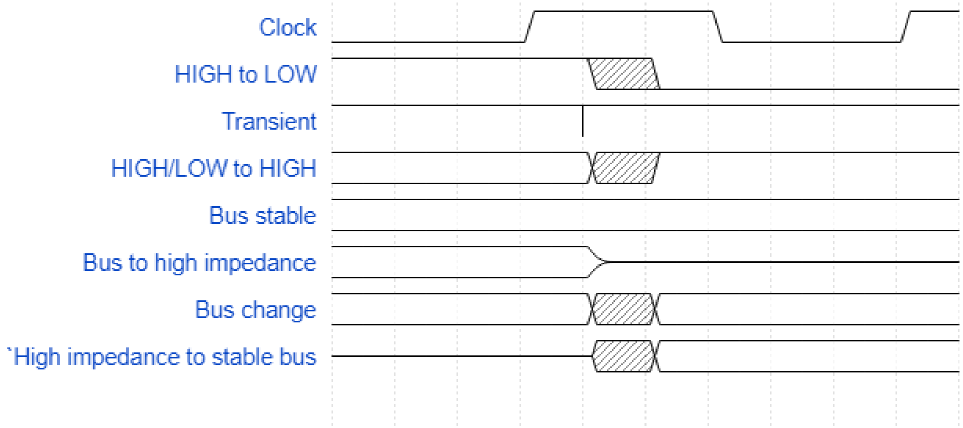
SMALL CAPITALS

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

2. *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 3.6. Therefore, several parameters for future purpose could be omitted.

3. *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.

3.1. 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 3-1: NUMBER OF COHERENT-AGENT INTERFACE UNITS(CAIU)

Number of CAIUs	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value			1	32	N/A
Constraints	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				

TABLE 3-2: NUMBER OF NON-COHERENT AGENT INTERFACE UNITS(NC-AIU)

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

TABLE 3-3: NUMBER OF DISTRIBUTED MEMORY INTERFACES

Number of DMIs	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	.
Value			1	16	N/A
Constraints	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				

TABLE 3-4: NUMBER OF SNOOP FILTERS

Number of SFs	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value			1	16	N/A
Constraints					
Customer Description	Number of Snoop Filters				
Engineering Description	Total number of Snoop Filters configured in an Ncore interconnect				

TABLE 3-5: NUMBER OF DISTRIBUTED VIRTUAL MEMORY SYSTEM ENGINES

Number of DVEs	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value			1	1	N/A
Constraints					
Customer Description	Number of DVEs				
Engineering Description	Total number of distributed virtual memory system engines configured in an Ncore interconnect				

TABLE 3-6: NUMBER OF DISTRIBUTED COHERENCY ENGINES

Number of DCEs	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value			1	16	N/A
Constraints					
Customer Description	Number of DCEs				
Engineering Description	Total number of distributed coherency engines configured in an Ncore Interconnect				

TABLE 3-7: NUMBER OF DISTRIBUTED IO INTERFACES

Number of DIIs	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value			1	16	N/A
Constraints					
Customer Description	Number of DIIs				
Engineering Description	Total number of distributed IO interfaces configured in an Ncore Interconnect				

4. *System User Settable Parameters*

4.1. Project name parameters

Table 4-1: projectName parameter

Name: projectName					Visibility: User
	Architecture		**Release**		**Default**
	String		*String*		
Value					
Constraints					
Customer Description	Project Name.				
Engineering Description					

4.2. System level connectivity parameters

As described in Section 3.4.5 Message connectivity and network mapping of NCore System Architecture specification version 0.76, 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 two 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 4-2: coherentTemplate Parameter

Name: coherentTemplate		Type: Enum	Visibiltiy: User
	Architecture	**Release**	**Default**
	Enum	*Enum*	
Value	TwoCtrlOneDataTemplate, ThreeCtrlOneDataTemplate, FourCtrlOneDataTemplate		FourCtrlOneDataTemplate
Constraints			
Customer Description	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			

New parameters are introduced to specify ports interleaving and to report connectivity information to AIUs.

Table 4-3: nAiuPorts Parameter

Name: nAiuPorts			Type: Int		Visibiltiy: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	16	1	8	1
Constraints	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 Figure 4-1, here it shows as a mutliported NCAIU with two AXI ports specified by nNativeInterfacePorts and then 2 NCAIUs specified by nAiuPorts.				

Table 4-4: aPrimaryAiuPortBits Parameter

Name: aPrimaryAiuPortBits		Type: Int	Visibiltiy: User
	Architecture	**Release**	**Default**
Value			array of integers
Constraints	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	

Table 4-5: aSecondaryAiuPortBits Parameter

Name: aSecondaryAiuPortBits		Type: Int	Visibiltiy: User
	Architecture	**Release**	**Default**
Value			Array of strings
Constraints	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		

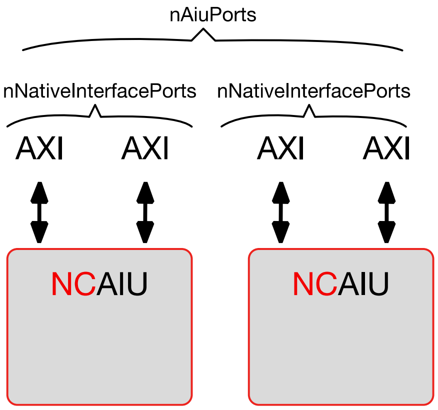


Figure 4-1 nAiuPorts Parameter

4.3. System address map parameters

Since NCore 3.0, we could have up to 24 configurable memory regions, and each memory region would be configured using registers. Please refer system architecture specification version 0.76 for the details (Section 5.4.7 Definition of a Memory Address Map).

Table 4-6: nGPRA parameter

Name: nGPRA			Type: int		Visibiltiy: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	2	24	2	24	2[1]
Constraints					
Customer Description	Number of general purpose address regions that the system can support				
Engineering Description					

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

Table 4-7: dceInterleavingPrimaryBits

Name: dceInterleavingPrimaryBits		Type: Int	Visibiltiy: User
	Architecture	**Release**	**Default**
	Array of Integers	*Array of Integers*	
Value			
Constraints			
Customer Description	System directory primary select bits. 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 DMIs to be accessed		
Engineering Description			

Table 4-8: dceInterleavingSecondaryBits

dceInterleavingSecondaryBits		Type: Int	Visibility: None
	Architecture	**Release**	**Default**
	Array of Integers	*Array of Integers*	
Value		Will not be released at NCore 3.2	
Constraints			
Customer Description	The secondary bits are chosen on oa 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	Not tested yet. Will not be released at NCore 3.2.		

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

4.4. 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 9 Functional Safety in the NCore System Architecture Specification version 0.76.

TABLE 4-9: RESILIENCE ON/OFF PARAMETER

Name: resilienceEnabled		Type: Boolean	Visibiltiy: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False		FALSE
Constraints			
Customer Description	Enable resilience-related features in the Ncore system.		
Engineering Description			

TABLE 4-10: SAFETYCONFIG PARAMETER

Name: safetyConfig			Visibility: User
	Architecture	**Release**	**Default**
	String	*String*	
Value	"NO_ASIL", "ASIL_A", "ASIL_B", "ASIL_D"	"NO_ASIL", "ASIL_A", "ASIL_B", "ASIL_D"	"NO_ASIL"
Constraints			
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.		

TABLE 4-11: DUPLICATION ENABLE PARAMETER

Name: duplicationEnabled		Type: Boolean	Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False		FALSE
Constraints			
Customer Description	Enable unit duplication for all Ncore units only. Memories and interconnect logic are not duplicated; they may be protected separately		
Engineering Description			

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

TABLE 4-12: NATIVE INTERFACE PROTECTION PARAMETER

Name: nativeIntfProtEnabled		Type: Boolean	Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False		FALSE
Constraints			
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			

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 4-13: INTERUNITDELAY PARAMETER

Name: interUnitDelay			Type: Int		Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	4	1	4	1
Constraints					
Customer Description	Delay between functional unit and delay unit. Delay can be specified in number of clock cycles.				
Engineering Description					

TABLE 4-14: RESILIENCYPROTECTIONTYPE PARAMETER

Name: resiliencyProtectionType			Visibility: User
	Architecture	**Release**	**Default**
	String	*String*	
Value	'"NONE"', '"PARITY"', '"SECDED"	'"NONE"', '"PARITY"', '"SECDED"	None
Constraints			
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.		

TABLE 4-15: MEMORYPROTECTIONTYPE PARAMETER

Name: memoryProtectionType			Visibility: User
	Architecture	**Release**	**Default**
	String	*String*	
Value	'"NONE"', '"PARITY"', '"SECDED"	'"NONE"', '"PARITY"', '"SECDED"	None
Constraints			
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.		

TABLE 4-16: FNDISABLERESILIENCYBISTDEBUGPIN PARAMETER

Name: fnDisableResiliencyBistDebugPin			Type: Int		Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	1	0	1	0
Constraints					
Customer Description	When set removes BIST and trace & debug disable pin.				
Engineering Description	When set removes BIST and trace & debug disable pin.				

4.4. 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 4-17: TIMEOUTTHRESHOLDPARAMETER

Name: timeoutThreshold			Visibility: User
	Architecture	**Release**	**Default**

	Min	*Max*	*Min*	*Max*	
Value	1	2147483647	1	2147483647	16384
Constraints					
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					

TABLE 4-18: MEMORYPROTECTIONTYPE PARAMETER

Name: memoryProtectionType			Visibility: User
	Architecture	**Release**	**Default**
	String	*String*	
Value	'"NONE"', '"PARITY"', '"SECDED"	'"NONE"', '"PARITY"', '"SECDED"	None
Constraints			
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			

4.5. System QoS parameters

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

TABLE 4-19: QOSENABLED PARAMETER

Name: qosEnabled		Visibility: User
	Architecture	**Default**
	Boolean	
Value	True, False	False
Constraints		
Customer Description	Enable QoS support	
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 value is being used for the QoS arbitration in skid buffer and OTT entries in AIU, DCE, and DMI.

TABLE 4-20: QOSMAP PARAMETER

Name: qosMap		Visibility: User

	Architecture	Release	Default
	List of String	*List of String*	
Value	"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"],
Constraints	Map 4-bit native interface QoS value to a 3-bit priority used inside an Ncore		
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	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],		

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.

Table 4-21: qosEventThreshold parameter

Name: qosEventThreshold					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	8192	1	8192	16
Constraints					
Customer Description	QoS starvation threshold. Maximum number of high priority requests that can bypass a lower priority request.				
Engineering Description					

4.6. 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 4-22: nMainTraceBufSize parameter

Name: nMainTraceBufSize					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	32	4096	32	1024	64
Constraints					
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.				

All AIUs in the NCore system including those that support trace signaling shall have the capability to initiate transaction tracing using internal CSRs. An 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

Table 4-23: nTraceRegisters parameter

Name: nTraceRegisters					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	8	1	4	1
Constraints	Minmum 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 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.

Table 4-24: nUnitTraceBufSizes parameter

Name: nMainTraceBufSize					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	8	32	8	16	8
Constraints	Allowable size are power of 2				
Customer Description	Number of trace entries in each Ncore Unit. Each entry is 64 bytes				
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.

TABLE 4-25: FNDEBUGAPBENABLE PARAMETER

Name: fnDebugAPBEnable					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	1	0	1	1
Constraints					
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	When set enables an APB slave port on the CSR network. This port is expected to be used for on chip debug purposes only.				

4.7. System level physical parameters

TABLE 4-26: SYNCDEPTH PARAMETER

Name: syncDepth			Visibility: User
	Architecture	**Release**	**Default**
	Valid values	*Valid values*	
Value	2, 3, 4	2,3,4	2
Constraints			
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		

4.8. System engineering parameters

Engineering Only parameter should not be visible to the customer.

Table 4-27: assertionEnable parameter

Name: assertionEnable				Visibility: Engg
	Description	**Type**	**Default**	**Notes**
assertionEnable	Enable HW assertions	Boolean	FALSE	Engineering Only

Table 4-28: EngVerId parameter

Name: EngVerId		Visibility: Engg
	Type	**Default**
	32 bits integer	
Value		
Constraints		
Customer Description		
Engineering Description	Refer to Engineering version id 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 xxxxxxxxxxxxxxxxxxx0001111101001, where x is a mpf hash.	

Table 4-29: ImplVerId parameter

Name: ImplVerId		Visibility: Engg
	Type	**Default**
	16 bits interger	
Value	Format: {4’d, 4’d, 4’d, 4’d}	
Constraints		
Customer Description		

Engineering Description	Refer to Engineering version id 16 bits to store Ncore version. 4bits per digit. For example Ncore 3.6.2.6 => {16'h3626} or { 4'd3, 4'd6, 4'd2, 4'd6 }

4.9. RISC-V related parameters

Table 4-30: NoDVM parameter

Name: noDVM				Visibility: User
	Description	**Type**	**Default**	**Notes**
noDVM	Disable DVM related functionality throughout an Ncore and does not reserve any DVM related credits such as DVM snoop credits across all Ncore units	Boolean	FALSE	

Please note that there isn't ANY DVM transaction that can be issued to an Ncore once noDVM is set true. Otherwise, unexpected behavior might happen.
Also, this parameter is mainly employed to optimize for configurations where exclusive RISC-V core usage for ALL processing elements connected to an Ncore.

4.10. RTL prefix addition parameter

Table 4-31: useRtlPrefix parameter

Name: useRtlPrefix				Visibility: User
	Description	**Type**	**Default**	**Notes**
useRtlPrefix	Once it is set as true, it will add a prefix globally to the entire Ncore units' module names	Boolean	FALSE	
Customer 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.			
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.			

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.

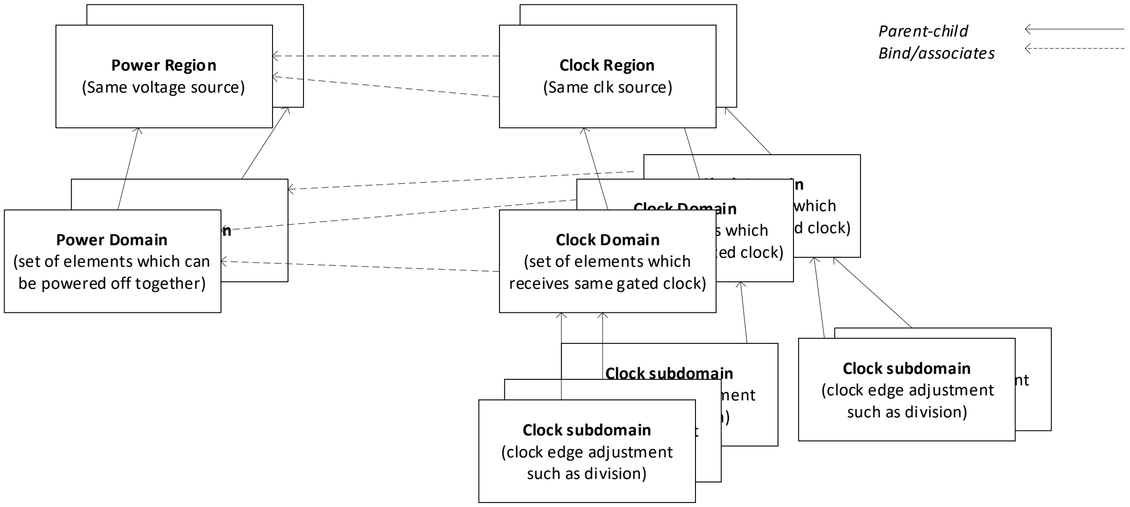

Power Region
(Same voltage source)
Clock Region
(Same clk source)
Parent-child
Bind/associates
Power Domain
(set of elements which can be powered off together)
Clock Domain
(set of elements which receives same gated clock)
Clock subdomain
(clock edge adjustment such as division)
Clock subdomain
(clock edge adjustment such as division)
Clock subdomain

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

<u>Ncore 3.6 supports three levels of clock gating:</u>

- 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.

How to achieve third-level clock gating scheme is under discussion https://jira.arteris.com/browse/MAES-3988. The below is proposal from John/MK.

At NCore 3.6, we would want to support (only) **NCore unit clock gating** using q-channel. To achieve this, the below updates would be required:

- By default, clock subdomains are async with other clock domains, but NCore 3.6 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 following changes would be needed:

- 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.

NCore 3.6 restrictions:

- NCore 3.6 supports only single power domain.
    - NCore 3.6 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 reponsiblity to support multiple power domain using clock region/domain capabiltiy.
- NCore 3.6 does NOT support retention mode.
- NCore 3.6 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

Name: Frequency (KHz)					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1,000 (1 MHz)	3,000,000 (3 GHz)	1,000 (1 MHz)	3,000,000 (3 GHz)	500,000 (500 MHz)
Constraints	The granularity of the frequency is in Kilo Hz. The working range is from 1 MHz to 3 GHz. Default at 500 MHz. Working frequency is very technology node dependent, which implies sometimes a frequency within the range still can't be achievable for a given technology node.				

Customer Description	The working range is from 1 MHz to 3 GHz. Default at 500 MHz. Working frequency is very technology node dependent, which implies sometimes a frequency within the range still can't be achievable for a given technology node.
Engineering Description	The working range is from 1 MHz to 3 GHz. Default at 500 MHz. Working frequency is very technology node dependent, which implies sometimes a frequency within the range still can't be achievable for a given technology node.

Table 5-2: Parameter related with **Clock Region**: unitClockGating

Name: unitClockGating			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid values*	
Value	True, False		FALSE
Constraints			
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.		

Table 5-3: Parameter related with **Clock Domain**: Gating

Name: Gating			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	always_on, external	always_on, external	always_on
Constraints			
Customer Description	Specify 'always_on' if no gating applied, or 'external' if gated control logic is applied externally		
Engineering Description			

6. *Memory Map User Settable Parameters*

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 address mapping 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

Name: memoryBase			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid values*	0x2e800000
Value			0x0
Constraints			
Customer Description	Specify CSR region base address. This address must be aligned to the size specified.		
Engineering Description			

Table 6-2: Parameter related with **CSR Region**: memorySize

Name: memorySize			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid values*	
Value	1MB	1MB	1MB
Constraints			
Customer Description	CSR sized is fixed as 1MB from NCore 3.2		
Engineering Description			

Table 6-3: Parameter related with **Boot Region**: memoryBase

Name: memoryBase			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid values*	
Value			0x0
Constraints			
Customer Description	Specify boot region base address. This address must be aligned to the size specified.		
Engineering Description	Must be aligned to 4KB		

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

Name: memorySize			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid values*	
Value	Min: 4KB, Max: 32KB	Min: 4KB, Max: 32KB	16K
Constraints			
Customer Description	Specifies the size of boot region. Minimum is 4KB and must be a power of two.		
Engineering Description			

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

Name: mg_ref			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid values*	
Value			
Constraints			
Customer Description	Specify the DMI interleave group associated with the boot region.		
Engineering Description	If mg_ref is specified, channel_ref cannot be specified		

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

Name: channel_ref			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid values*	
Value			
Constraints			
Customer Description	Specify the DII group associated with the boot region.		
Engineering Description	If channel_ref is specified, mg_ref cannot be specified		

DYNAMIC MEMORY GROUP:

Dynamic memory group is to define GPARS (Number of GPARS is configured by Table 4-6: nGPRA parameter). 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 3.3 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

Name: primaryInterleavingBitOne			Type: Int		Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	8192	0	8192	0
Constraints					
Customer Description					
Engineering Description	This primaryInterleavingBitOne is per MIF.				

Table 6-8: primaryInerleavingBitTwo

Name: primaryInterleavingBitTwo			Type: Int		Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	8192	0	8192	0
Constraints					
Customer Description					
Engineering Description	This primaryInterleavingBitTwo is per MIF.				

Table 6-9: primaryInerleavingBitThree

Name: primaryInterleavingBitThree			Type: Int		Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	8192	0	8192	0
Constraints					
Customer Description					
Engineering Description	This primaryInterleavingBitThree is per MIF.				

Table 6-10: primaryInerleavingBitFour

Name: primaryInterleavingBitFour		Type: Int	Visibility: User
	Architecture	**Release**	**Default**
	Min	*Max*	*Min*
Value	0	8192	
Constraints			
Customer Description			
Engineering Description	This primaryInterleavingBitFour is per MIF.		

7. Socket User Settable Parameters

7.1. Native interface parameters

TABLE 7-1: FNNATIVEINTERFACE PARAMETER

Name: fnNativeInterface		Type: Enum	Visibility: user Settable
	Architecture	**Release**	**Default**
	Valid Values	Valid Values	
Value	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			
Customer Description	Selects native interface type for a CAIU		
Engineering Description	Selects native interface type CHIA is deprecated in 3.6 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		

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 20.8.

NCore 3.6 restrictions:

- AwID and ArID need to be restricted to be same for an interface irrespective of it being a master or slave.
- 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.

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

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)

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

Name: wArID					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	20/32	1	20/32	6
Constraints					
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.				

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

[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).

Name: wAwID					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	20/32	1	20/32	6
Constraints					
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.				

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

Name: wAddr					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	12	64	12	64	32
Constraints					
Customer Description	Specify the width of AXI4 interface address bits.				
Engineering Description					

Table 7-5: Parameter related with AXI4 Interface: wData

Name: wData			Visibility: User
	Architecture	**Release**	**Default**
	Valid values	*Valid values*	
Value	[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
Constraints			
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			

Table 7-6: Parameter related with AXI4 Interface: AwUser

Name: wAwUser					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	32	0
Constraints					
Customer Description	Width of user bit on AW AXI4 Interface				
Engineering Description					

Table 7-7: Parameter related with AXI4 Interface: ArUser

Name: wArUser					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	32	0
Constraints					
Customer Description	Width of user bit on AR AXI4 Interface				
Engineering Description					

Table 7-8: Exclusive_Accesses Property for AXI4 Interface

Name: exclusiveAccesses					Visibility: Engg
	Architecture		**Release**		**Default**
	Boolean		*Boolean*		
Value	True, False		True, False		True
Constraints	It is always turned on; Exclusive accesses are supported by default				
Customer Description	It is always turned on; Exclusive accesses are supported by default				
Engineering Description	It is always turned on; Exclusive accesses are supported by default				

Table 7-9: Coherency_Connection_Signals Property parameter For AXI4 interface

Name: useSysCoInt					Visibility: Engg
	Architecture		**Release**		**Default**
	Boolean		*Boolean*		
Value	*True, False*		*True, False*		False
Constraints	Shall be set to False always. The SysCo handshake is always carried out via the usage of CSR.				
Customer Description	Shall be set to False to an AXI4 interface-based agent no matter hasProxyCache parameter defined in Table 14-10 is set as true or false. The SysCo handshake is always carried out via the usage of CSR.				
Engineering Description	Shall be set to False to an AXI4 interface-based agent no matter hasProxyCache parameter defined in Table 14-10 is set as true or false. The SysCo handshake is always carried out via the usage of CSR.				

7.3. AXI5 Interface

Table 7-10: Parameter related with AXI5 Interface: wArID

Name: wArID					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	20/32	1	20/32	6
Constraints					
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.				

Table 7-11: Parameter related with AXI5 Interface: wAwID

Name: wAwID					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	20/32	1	20/32	6
Constraints					
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.				

Table 7-12: Parameter related with AXI5 Interface: wAddr

Name: wAddr					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	12	64	12	64	32
Constraints					
Customer Description	Specify the width of AXI5 interface address bits.				
Engineering Description					

Table 7-13: Parameter related with AXI5 Interface: wData

Name: wData					Visibility: User
	Architecture		**Release**		**Default**

	Valid values	*Valid values*	
Value	[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
Constraints			
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			

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

Name: wAwUser					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	32	0
Constraints					
Customer Description	Width of user bit on AW AXI5 Interface				
Engineering Description					

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

Name: wArUser					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	32	0
Constraints					
Customer Description	Width of user bit on AR AXI5 Interface				
Engineering Description					

TABLE 7-16: EXCLUSIVE_ACCESSES PROPERTY FOR AXI5 INTERFACE

Name: exclusiveAccesses			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	It is always turned on; Exclusive accesses are supported by default		
Customer Description	It is always turned on; Exclusive accesses are supported by default		
Engineering Description	It is always turned on; Exclusive accesses are supported by default		

TABLE 7-17: CHECK_TYPE PROPERTY FOR AXI5 INTERFACE

Name: checkType			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	*NONE, ODD_PARITY_BYTE_ALL*	*NONE, ODD_PARITY_BYTE_ALL*	NONE
Constraints	Odd parity checking included for all signals. Each bit of the parity signal generally covers up to 8 bits.		
Customer Description	Odd parity checking included for all signals. Each bit of the parity signal generally covers up to 8 bits.		
Engineering Description	Odd parity checking included for all signals. Each bit of the parity signal generally covers up to 8 bits.		

Note: 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.

TABLE 7-18: ATOMIC_TRANSACTIONS PROPERTY FOR AXI5 INTERFACE

Name: atomicTransactions			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	False
Constraints	Once this property is enabled, atomic transactions are supported for the AXI5 interface. 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 AXI5 interface		
Engineering Description	Once this property is enabled, atomic transactions are supported for the AXI5 interface		

Note: Once 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.

TABLE 7-19: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR AXI5 INTERFACE

Name: useSysCoInt			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	False

Constraints	Always set as False to an AXI5 interface-based CPU and it takes effect when hasProxyCahce parameter defined in Table 14-10 is set as true.
Customer Description	Always set as False to an AXI5 interface-based CPU and it takes effect when hasProxyCahce parameter defined in Table 14-10 is set as true.
Engineering Description	Always set as False to an AXI5 interface-based CPU and it takes effect when hasProxyCahce parameter defined in Table 14-10 is set as true.

Table 7-20: Trace_Signals Property for AXI5 Interface

Name: eTrace			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Always set as true for AXI5 interface		
Customer Description	Always set as true for AXI5 interface		
Engineering Description	Once it is turned on, the relevant *TRACE signals will be added across different channels for the AXI5 interface.		

7.4. ACE Interface

Table 7-21: Parameter related with ACE Interface: wArID

Name: wArID					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	20	1	20	6
Constraints					
Customer Description	Specify the ArId width of ACE interface .				
Engineering Description					

Table 7-22: Parameter related with ACE Interface: wAwID

Name: wAwID					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	20	1	20	6
Constraints					
Customer Description	Specify the AwId width of ACE interface .				
Engineering Description					

Table 7-23: Parameter related with ACE Interface: wAddr

Name: wAddr			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	

Value	32, 40, 44, 48	32, 40, 44, 48	32
Constraints			
Customer Description	Specify the width of ACE interface address bits. ACE only: 32, 40, 44, 48 ACE with CHI: 44, 48		
Engineering Description			

Table 7-24: Parameter related with ACE Interface: wData

Name: wData			Visibility: User
	Architecture	**Release**	**Default**
	Valid values	*Valid values*	
Value	64, 128, 256	64, 128, 256	64
Constraints			
Customer Description	Specify the width of ACE interface data bits		
Engineering Description			

Table 7-25: Parameter related with ACE Interface: AwUser

Name: wAwUser					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	32	0
Constraints					
Customer Description	Width of AwUser bit on AW channel of ACE Interface				
Engineering Description					

Table 7-26: Parameter related with ACE Interface: ArUser

Name: wArUser					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	32	0
Constraints					
Customer Description	Width of ArUser bit on AR channel of ACE Interface				
Engineering Description					

Table 7-27: Exclusive_Accesses Property for ACE Interface

Name: exclusiveAccesses			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	

Value	True, False	True, False	True
Constraints	It is always turned on; Exclusive accesses are supported by default		
Customer Description	It is always turned on; Exclusive accesses are supported by default		
Engineering Description	It is always turned on; Exclusive accesses are supported by default		

TABLE 7-28: PARAMETER RELATED TO EVENTIN INTERFACE

Name: useEventInInt			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	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.		
Customer Description	Setting the parameter to enable the connection of EventInReq and EventInAck interface to the AIU. if customer's CPU does not have the interface, and does not set False to the parameter, it is recommended typing EventInAck to EventInReq to zerios.		
Engineering Description	Connect the I/O to the SysReq Receiver hardware.		

TABLE 7-29: PARAMETER RELATED TO EVENTOUT INTERFACE

Name: useEventOutInt			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	Default True to ACE interface CPU, but if customer's CPU does not have the EventOut interface, customers can set the parameter to be False.		
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 tying EventOutReq to 0.		
Engineering Description	Connect the I/O to the SysReq Sender hardware.		

TABLE 7-30: SHAREABLE_TRANSACTIONS PROPERTY FOR ACE INTERFACE

Name: shareableTransactions			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Fixed as always True, AxDOMAIN signals will present in ACE interface, inner and outer shareable transactions are supported		
Customer Description	Fixed as always True, AxDOMAIN signals will present in ACE interface, inner and outer shareable transactions are supported		
Engineering Description	Fixed as always True, AxDOMAIN signals will present in ACE interface, inner and outer shareable transactions are supported		

Table 7-31: Continuous_Cache_Line_Read_Data Property for ACE Interface

Name: continuousCacheLineReadData			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Once it is turned on, continuous cache line data return is supported for the ACE interface		
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		

Table 7-32: DVM_v8 Property for ACE Interface

Name: dvmV8			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Once it is turned on, DVM version 8.0 functionalities are supported for the ACE interface		
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		

Table 7-33: DVM_v8.1 Property for ACE Interface

Name: dvmV8.1			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Once it is turned on, DVM version 8.1 functionalities are supported for the ACE interface. 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.		

Table 7-34: DVM_Message_Support Property for ACE Interface

Name: dvmMessageSupport			Visibility: Engg
	Architecture	**Release**	**Default**
	Valid values	*Valid values*	

Value	Bidirectional, Receiver, False	Bidirectional, Receiver, False	Bidirectional
Constraints	Currently, bidirectional is always supported in Ncore so this property is fixed and the customer can not change it		
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		

TABLE 7-35: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR ACE INTERFACE

Name: useSysCoInt			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	False
Constraints	Always set as False to an ACE interface-based agent. The SysCo handshake is always carried out with the usage of CSR. This is not a property required by ARM but is required for Ncore.		
Customer Description	Always set as False to an ACE interface-based agent. The SysCo handshake is always carried out with the usage of CSR. This is not a property required by ARM but is required for Ncore.		
Engineering Description	Always set as False to an ACE interface-based agent. The SysCo handshake is always carried out with the usage of CSR. This is not a property required by ARM but is required for Ncore.		

7.5. ACE5 Interface

TABLE 7-36: PARAMETER RELATED WITH ACE5 INTERFACE: WARID

Name: wArID					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	20	1	20	6
Constraints					
Customer Description	Specify the ArId width of ACE5 interface .				
Engineering Description					

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

Name: wAwID					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	20	1	20	6
Constraints					
Customer Description	Specify the AwId width of ACE5 interface.				
Engineering Description					

Table 7-38: Parameter related with ACE5 Interface: wAddr

Name: wAddr			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	32, 40, 44, 48	32, 40, 44, 48	32
Constraints			
Customer Description	Specify the width of ACE5 interface address bits. ACE only: 32, 40, 44, 48 ACE with CHI: 44, 48		
Engineering Description			

Table 7-39: Parameter related with ACE5 Interface: wData

Name: wData			Visibility: User
	Architecture	**Release**	**Default**
	Valid values	*Valid values*	
Value	64, 128, 256	64, 128, 256	64
Constraints			
Customer Description	Specify the width of ACE5 interface data bits		
Engineering Description			

Table 7-40: Parameter related with ACE5 Interface: AwUser

Name: wAwUser					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	32	0
Constraints					
Customer Description	Width of AwUser bit on AW channel of ACE5 Interface				
Engineering Description					

Table 7-41: Parameter related with ACE5 Interface: ArUser

Name: wArUser					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	32	0
Constraints					
Customer Description	Width of ArUser bit on AR channel of ACE5 Interface				
Engineering Description					

TABLE 7-42: COHERENCY_CONNECTION_SIGNALS PROPERTY PARAMETER FOR ACE5 INTERFACE

Name: useSysCoInt			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	False
Constraints	Always set as false to an ACE5 interface-based CPU. If customer's CPU does have the SysCo interface, customers can set the parameter to be true.		
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 to Ncore if it is set as True.		

TABLE 7-43: PARAMETER RELATED TO EVENTIN INTERFACE

Name: useEventInInt			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	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.		
Customer Description	Setting the parameter to enable the connection of EventInReq and EventInAck 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 EventInAck to EventInReq.		
Engineering Description	Connect the I/O to the SysReq Receiver hardware.		

TABLE 7-44: PARAMETER RELATED TO EVENTOUT INTERFACE

Name: useEventOutInt			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	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.		
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.		

TABLE 7-45: CHECK_TYPE PROPERTY FOR ACE5 INTERFACE

Name: checkType			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	

Value	*NONE,* *ODD_PARITY_BYTE_ALL*	*NONE,* *ODD_PARITY_BYTE_ALL*	NONE
Constraints	Odd parity checking included for all signals. Each bit of the parity signal generally covers up to 8 bits.		
Customer Description	Odd parity checking included for all signals. Each bit of the parity signal generally covers up to 8 bits.		
Engineering Description	Odd parity checking included for all signals. Each bit of the parity signal generally covers up to 8 bits.		

Note: 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.

Table 7-46: Exclusive_Accesses Property for ACE5 Interface

Name: exclusiveAccesses			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	It is always turned on; Exclusive accesses are supported by default		
Customer Description	It is always turned on; Exclusive accesses are supported by default		
Engineering Description	It is always turned on; Exclusive accesses are supported by default		

Note: 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.

Table 7-47: Shareable_Transactions Property for ACE5 Interface

Name: shareableTransactions			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Fixed as always True, AxDOMAIN signals will present in ACE5 interface, inner and outer shareable transactions are supported		
Customer Description	Fixed as always True, AxDOMAIN signals will present in ACE5 interface, inner and outer shareable transactions are supported		
Engineering Description	Fixed as always True, AxDOMAIN signals will present in ACE5 interface, inner and outer shareable transactions are supported		

Table 7-48: Continuous_Cache_Line_Read_Data Property for ACE5 Interface

Name: continuousCacheLineReadData			Visibility: User
	Architecture	**Release**	**Default**

	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Once it is turned on, continuous cache line data return is supported for the ACE5 interface		
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		

Table 7-49: DVM_v8 Property for ACE5 Interface

Name: dvmV8			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Once it is turned on, DVM version 8.0 functionalities are supported for the ACE5 interface		
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		

Table 7-50: DVM_v8.1 Property for ACE5 Interface

Name: dvmV8			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Once it is turned on, DVM version 8.1 functionalities are supported for the ACE5 interface. 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.		

Table 7-51: DVM_Message_Support Property for ACE5 Interface

Name: dvmMessageSupport			Visibility: Engg
	Architecture	**Release**	**Default**
	Valid values	*Valid values*	
Value	Bidirectional, Receiver, False	Bidirectional, Receiver, False	Bidirectional
Constraints	Currently, bidirectional is always supported in Ncore so this property is fixed and the customer can not change it		
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		

Table 7-52: Trace_Signals Property for ACE5 Interface

Name: eTrace		Visibility: Engg	
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Always set as true for ACE5 interface		
Customer Description	Always set as true for ACE5 interface		
Engineering Description	Once it is turned on, the relevant *TRACE signals will be added across different channels for the ACE5 interface.		

7.6. ACE5-LITE Interface

Table 7-53: Parameter related with ACE5-LITE/ACE5-LiteDVM Interface: wArID

Name: wArID					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	20	1	20	6
Constraints					
Customer Description	Specify the ArID width of ACE5-Lite/ACE5-LiteDVM interface.				
Engineering Description					

Table 7-54: Parameter related with ACE5-LITE/ACE5-LiteDVM Interface: wAwID

Name: wAwID					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	20	1	20	6
Constraints					
Customer Description	Specify the width of ACE5-Lite/ACE5-LiteDVM interface AwId bits.				
Engineering Description					

Table 7-55: Parameter related with ACE5-LITE/ACE5-LiteDVM Interface: wAddr

Name: wAddr			Visibility: User
	Architecture	**Release**	**Default**

	Min	*Max*	*Min*	*Max*	
Value	12	64	12	64	32
Constraints					
Customer Description	Specify the width of ACE5-Lite/ACE5-LiteDVM interface Address bits. The address width needs to be equal to 44 or bigger once DVM is enabled.				
Engineering Description	Specify the width of ACE5-Lite/ACE5-LiteDVM interface Address bits. The address width needs to be equal to 44 or bigger once DVM is enabled.				

Table 7-56: Parameter related with ACE5-LITE/ACE5-LiteDVM Interface: wData

Name: wData					Visibility: User
	Architecture		**Release**		**Default**
	Valid values		*Valid values*		
Value	64, 128, 256		64, 128, 256		64
Constraints					
Customer Description	Specify the width of ACE5-Lite/ACE5-LiteDVM interface data bits				
Engineering Description					

Table 7-57: Parameter related with ACE5-LITE/ACE5-LiteDVM Interface: AwUser

Name: wAwUser					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	32	0
Constraints					
Customer Description	Width of user bit on AW ACE5-Lite/ACE5-LiteDVM Interface				
Engineering Description					

Table 7-58: Parameter related with ACE5-LITE/ACE5-LiteDVM Interface: ArUser

Name: wArUser					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	32	0
Constraints					
Customer Description	Width of user bit on AR ACE5-LITE/ACE5-LiteDVM Interface				
Engineering Description					

Table 7-59: Parameter to Enable DVM Functionalities for the ACE5-LITE Interface: DVM

Name: **DVM**					Visibility: User
	Architecture		**Release**		**Default**
	Valid values		*Valid values*		
Value	True, False		True, False		False
Constraints	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. Once it is set as true, the properties such as DVM_v8, DVM_v8.1, DVM_v8.4, and DVM_Message_Support properties start to take effect. If it is set as FALSE, DVM_v8, DVM_v8.1 and DVM_v8.4, and DVM_Message_Support should NOT be available for selection. Once any of the three DVM property is enabled, a Maestro check to ACE5-liteDVM address width needs to be applied in Table 7-55.
Engineering Description	Set this parameter true will make ACE5-Lite interface to be an ACE5-LiteDVM interface.

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

Name: dvmV8			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	False
Constraints	Only available for selection when DVM parameter defined in Table 7-59 is set as TRUE. Once it is turned on, DVM version 8.0 functionalities are supported for the ACE5-LiteDVM interface		
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		

TABLE 7-61: DVM_V8.1 PROPERTY FOR ACE5-LITEDVM INTERFACE

Name: dvmV8.1			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Only available for selection when DVM parameter defined in Table 7-59 is set as TRUE. 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.		
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.		

TABLE 7-62: DVM_V8.4 PROPERTY FOR ACE5-LITEDVM INTERFACE

Name: dvmV8.4			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	False

Constraints	Only available for selection when DVM parameter defined in Table 7-59 is set as TRUE. 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.
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.

Table 7-63: DVM_Message_Support Property for ACE5-LiteDVM Interface

Name: dvmMessageSupport			Visibility: Engg
	Architecture	**Release**	**Default**
	Valid values	*Valid values*	
Value	Bidirectional, Receiver, False	Bidirectional, Receiver, False	Bidirectional
Constraints	Only take effect when DVM parameter defined in Table 7-59 is set as TRUE. Currently, bidirectional is always supported in Ncore so this property is fixed and the customer can not change it		
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		

Table 7-64: Parameter related to EventOut Interface

Name: useEventOutInt			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	False
Constraints	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)		
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.		

Table 7-65: Parameter to enable the protection on the ACE5-Lite/ACE5-LiteDVM interface

Name: checkType			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	*NONE, ODD_PARITY_BYTE_ALL*	*NONE, ODD_PARITY_BYTE_ALL*	NONE

Constraints	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
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.

Note: 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.

Table 7-66: Coherency_Connection_Signals Property parameter For ACE5-LiteDVM interface

Name: useSysCoInt			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	False
Constraints	Always set as false to an ACE5-LiteDVM interface-based CPU. This parameter can only be selectable with DVM parameter defined in Table 7-59 is enabled, and it will be default as false under such a condition. If customer's CPU does have the SysCo interface, customers can set the parameter to be true.		
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 to an Ncore if it is set as True.		

Table 7-67: Atomic_Transactions Property for ACE5-Lite/ACE5-LiteDVM Interface

Name: atomicTransactions			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	False
Constraints	Once this property is enabled, atomic transactions are supported for the ACE5-Lite/ACE5-LiteDVM interface. 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		

Note: Once this property is enabled, extra relevant signals need to be added to the ACE5-Lite/ACE5-LiteDVM interface across all channels. Please refer to AMBA AXI and ACE Protocol Specification (version ARM IHI 0022H.c ID012621) for the complete signal list.

TABLE 7-68: EXCLUSIVE_ACCESSES PROPERTY FOR ACE5-LITE/ACE5-LITEDVM INTERFACE

Name: exclusiveAccesses			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	It is always turned on; Exclusive accesses are supported by default		
Customer Description	It is always turned on; Exclusive accesses are supported by default		
Engineering Description	It is always turned on; Exclusive accesses are supported by default		

TABLE 7-69: SHAREABLE_TRANSACTIONS PROPERTY FOR ACE5-LITE/ACE5-LITEDVM INTERFACE

Name: shareableTransactions			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Fixed as always True, AxDOMAIN signals will present in ACE5-Lite/ACE5-LiteDVM interface, inner and outer shareable transactions are supported		
Customer Description	Fixed as always True, AxDOMAIN signals will present in ACE5-Lite/ACE5-LiteDVM interface, inner and outer shareable transactions are supported		
Engineering Description	Fixed as always True, AxDOMAIN signals will present in ACE5-Lite/ACE5-LiteDVM interface, inner and outer shareable transactions are supported		

TABLE 7-70: CACHE_STASH_TRANSACTIONS PROPERTY FOR ACE5-LITE INTERFACE

Name: CacheStashTransactions			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Always set as true for ACE5_Lite interface		
Customer Description	Always set as true for ACE5_Lite interface		
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 in Table 14-26.		

TABLE 7-71: TRACE_SIGNALS PROPERTY FOR ACE5-LITE INTERFACE

Name: eTrace			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Always set as true for ACE5_Lite interface		
Customer Description	Always set as true for ACE5_Lite interface		

Engineering Description	Once it is turned on, the relevant *TRACE signals 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 enabled via what it is defined in Table 7-59.

7.7. ACE-LITE Interface

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

Name: wArID					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	20	1	20	6
Constraints					
Customer Description	Specify the ArID width of ACE-LITE interface.				
Engineering Description					

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

Name: wAwID					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	20	1	20	6
Constraints					
Customer Description	Specify the AwId width of ACE-LITE interface.				
Engineering Description					

TABLE 7-74: PARAMETER RELATED WITH ACE-LITE INTERFACE: WADDR

Name: wAddr					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	12	64	12	64	32
Constraints					
Customer Description	Specify the address width of ACE-LITE interface .				
Engineering Description					

TABLE 7-75: PARAMETER RELATED WITH ACE-LITE INTERFACE: WDATA

Name: wData			Visibility: User
	Architecture	**Release**	**Default**
	Valid values	*Valid values*	
Value	64, 128, 256	64, 128, 256	64
Constraints			
Customer Description	Specify the data width of ACE-LITE interface		

Engineering Description	

Table 7-76: Parameter related with ACE-LITE Interface: AwUser

Name: wAwUser					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	32	0
Constraints					
Customer Description	Width of AwUser bits on AW channel of ACE-LITE Interface				
Engineering Description					

Table 7-77: Parameter related with ACE-LITE Interface: ArUser

Name: wArUser					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	32	0
Constraints					
Customer Description	Width of user bit on ArUser bits on AR channel of ACE-LITE Interface				
Engineering Description					

Table 7-78: Parameter related with ACE-LITE Interface: eAC

Name: eAC			Visibility: No
	Architecture	**Release**	**Default**
	Valid values	*Valid values*	
Value	0, 1	0, 1	0
Constraints			
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.		

Table 7-79: Coherency_Connection_Signals Property parameter for ACE-Lite interface

Name: useSysCoInt			Visibility: No
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	False
Constraints	Shall be set to False to an ACE-Lite interface-based agent. The SysCo handshake is always done via the usage of CSR.		
Customer Description	Shall be set to False to an ACE-Lite interface-based agent. The SysCo handshake is always done via the usage of CSR.		

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.

Table 7-80: Shareable_Transactions Property for ACE-Lite Interface

Name: shareableTransactions			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	Fixed as always True, AxDOMAIN signals will present in ACE-Lite interface, inner and outer shareable transactions are supported		
Customer Description	Fixed as always True, AxDOMAIN signals will present in ACE-Lite interface, inner and outer shareable transactions are supported		
Engineering Description	Fixed as always True, AxDOMAIN signals will present in ACE-Lite interface, inner and outer shareable transactions are supported		

Table 7-81: Exclusive_Accesses Property for ACE-Lite Interface

Name: exclusiveAccesses			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	True
Constraints	It is always turned on; Exclusive accesses are supported by default		
Customer Description	It is always turned on; Exclusive accesses are supported by default		
Engineering Description	It is always turned on; Exclusive accesses are supported by default		

7.8. CHI Issue B Interface

Table 7-82: Parameter related with CHI_B_Interface: NodeID_Width

Name: NodeID_Width					Visibility: User
	Architecture		**Release**		**Default**
	Min	Max	min	max	
Value	7	11	7	11	7
Constraints					
Customer Description	Width of the Node ID of the CHI Interface.				
Engineering Description					

Table 7-83: Parameter related with CHI_B_Interface: wAddr

Name: wAddr		Visibility: User

	Architecture		Release		Default
	Min	Max	min	max	
Value	44	52	44	52	48
Constraints					
Customer Description	Width of the address on CHI interface.				
Engineering Description					

Table 7-84: Parameter related with CHI_B_Interface: REQ_RSVDC

Name: REQ_RSVDC			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	['0', '4', ‘8’,'12', '16', '24', '32']	['0', '4', ‘8’,'12', '16', '24', '32']	0
Constraints			
Customer Description	REQ_RSVDC is to define user-bit for command channel. Do not support user bit on data channel.		
Engineering Description			

Table 7-85: Parameter related with CHI_B_Interface: wData

Name: wData			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	128, 256	128, 256	128
Constraints			
Customer Description	Width of data on the Chi interface		
Engineering Description			

Table 7-86: Parameter related with CHI_B_Interface: enPoison

Name: enPoison			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	True, False	True, False	False
Constraints			
Customer Description	Enable Poison Bit		
Engineering Description			

Table 7-87: parameter related to sysco interface of CHI-B

Name: useSysCoInt			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	Always True to CHI-B interface CPU.		
Customer Description	Setting the parameter to enable the connection of SysCoReq and SysCoAck interface to the AIU. if customer's CPU doesn't use the interface, SysCoReq should be tied to 0.		
Engineering Description	It connects the corresponding I/O to the SysCo Engine hardware.		

TABLE 7-88: PARAMETER RELATED TO EVENTIN INTERFACE

Name: useEventInInt			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	Default True to CHI-B interface CPU, but if customer's CPU does not have the EventIn interface, customers can set the parameter to be False.		
Customer Description	Setting the parameter to enable the connection of EventInReq and EventInAck 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 EventInAck to EventInReq.		
Engineering Description	Connect the I/O to the SysReq Receiver hardware.		

TABLE 7-89: PARAMETER RELATED TO EVENTOUT INTERFACE

Name: useEventOutInt			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	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.		
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.		

TABLE 7-90: PARAMETER TO ENABLE THE PROTECTION ON THE CHI-B INTERFACE

Name: checkType			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	*NONE, ODD_PARITY_BYTE_ALL*	*NONE, ODD_PARITY_BYTE_ALL*	NONE

Constraints	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	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
Engineering Description	Necessary odd parity signals across ALL channels will be added once it is set as ODD_PARITY_BYTE_ALL.

7.9. CHI Issue E Interface

Table 7-91: Parameter related with CHI_E_Interface: NodeID_Width

Name: NodeID_Width					Visibility: User
	Architecture		**Release**		**Default**
	Min	Max	min	max	
Value	7	11	7	11	7
Constraints					
Customer Description	Width of the Node ID of the CHI Interface.				
Engineering Description					

Table 7-92: Parameter related with CHI_E_Interface: wAddr

Name: wAddr					Visibility: User
	Architecture		Release		Default
	Min	Max	min	max	
Value	44	52	44	52	48
Constraints					
Customer Description	Width of the address on CHI interface.				
Engineering Description					

Table 7-93: Parameter related with CHI_E_Interface: REQ_RSVDC

Name: REQ_RSVDC			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	['0', '4', ‘8’,'12', '16', '24', '32']	['0', '4', ‘8’,'12', '16', '24', '32']	0
Constraints			

Customer Description	REQ_RSVDC is to define user-bit for command channel. Do not support user bit on data channel.
Engineering Description	

TABLE 7-94: PARAMETER RELATED WITH CHI_E_INTERFACE: WDATA

Name: wData			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	128, 256	128, 256	128
Constraints			
Customer Description	Width of data on the Chi interface		
Engineering Description			

TABLE 7-95: PARAMETER RELATED WITH CHI_E_INTERFACE: ENPOISON

Name: enPoison			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	True, False	True, False	False
Constraints			
Customer Description	Enable Poison Bit		
Engineering Description			

TABLE 7-96: PARAMETER RELATED TO SYSCO INTERFACE OF CHI-E

Name: useSysCoInt			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	Always True to CHI-E interface CPU.		
Customer Description	It enables the connection of SysCoReq and SysCoAck interface to the AIU. if customer's CPU doesn't use the interface, SysCoReq should be tied to 0.		
Engineering Description	It connects the corresponding I/O to the SysCo Engine hardware.		

TABLE 7-97: PARAMETER RELATED TO EVENTIN INTERFACE

Name: useEventInInt			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	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.		
Customer Description	Setting the parameter to enable the connection of EventInReq and EventInAck 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 EventInAck to EventInReq.
Engineering Description	Connect the I/O to the SysReq Receiver hardware.

TABLE 7-98: PARAMETER RELATED TO EVENTOUT INTERFACE

Name: useEventOutInt			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	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.		
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.		

TABLE 7-99: PARAMETER TO ENABLE THE PROTECTION ON THE CHI-E INTERFACE

Name: checkType			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	*NONE, ODD_PARITY_BYTE_ALL*	*NONE, ODD_PARITY_BYTE_ALL*	NONE
Constraints	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	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		
Engineering Description	Necessary odd parity signals across ALL channels will be added once it is set as ODD_PARITY_BYTE_ALL.		

8. Concerto User Settable Parameters

All Concerto parameters are derived parameters.

9. *CAIU User Settable Parameters*

9.1. CAIU resource parameters

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.

TABLE 9-1: NOTTCTRLENTRIES FOR CAIU

Name: nOttCtrlEntries					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	4	128	8	128	8
Constraints	Applied to CHI-AIU and IOAIU				
Customer Description	Specify the maximum number of outstanding native transactions this AIU should support.				
Engineering Description					

TABLE 9-2: GENERIC PORTS PARAMETER FOR CAIU

Name: genericports			Visibility: User
	Architecture	Release	Default
Constraints	Applied to CHI-AIU and IOAIU		
Engineering Description	To assign user defined ports for place holder definition(Resiliency); Described in Chapter 21.1		

TABLE 9-3: MEMORY PARAMETER FOR CAIU

Name: Memory				Visibility: User
	Architecture	Release		Default
Constraints	Applied Only to IOAIU			
Engineering Description	This parameter is to assign SRAM. For the memory setting, refer Table 21-5. This is only for ACE.			

9.2. CAIU credit parameters

Specify the maximum number of CHI link credits this AIU will support. 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.

TABLE 9-4: NNATIVECREDITS PARAMETER FOR CAIU

nNativeCredits	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	2	15	2	15	2
Constraints	Applied Only to CHI-AIU				
Customer Description	Specify the maximum number of CHI link credits this AIU should support.				
Engineering Description					

Specify the maximum number of credits for coherent transactions per DCE.This should be determined based on required bandwidth and netwrok round trip latency.

TABLE 9-5: NDCECMDCREDITS FOR CAIU

nDceCmdCredits	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	2	16	2	16	2
Constraints	Applied to CHI-AIU and IOAIU				
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					

Specify the maximum number of credits for non-coherent transactions per DMI.This should be determined based on required bandwidth and netwrok round trip latency.

TABLE 9-6: NDMICMDCREDITS FOR CAIU

nDmiCmdCredits	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	2	16	2	16	2
Constraints	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					

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.

TABLE 9-7: NDIICMDCREDITS FOR CAIU

nDiiCmdCredits	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	2	16	2	16	2
Constraints	Applied to CHI-AIU and IOAIU				
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					

Specify the maximum number of outstanding stash snoops this AIU should support.

TABLE 9-8: NSTASHSNPCREDITS FOR CAIU

nStashSnpCredits	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	1	8	1	8	2
Constraints	Applied Only to CHI-AIU				
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				

9.3. CAIU address map parameter

TABLE 9-9: FNCSRACCESS_PARAMETER

fnCsrAccess	Architecture	Release	Default
	Valid Values	*Valid values*	
Value	True, False	True, False	True
Constraints	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.		

9.4. CAIU snoop filter parameters

TABLE 9-10: SNOOPFILTER_REF PARAMETER FOR CAIUS

SnoopFilter_Ref	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	0	64	0	64	0
Constraints	Applied to CHI-AIU and IOAIU.				
Customer Description	Specify the snoop filter associated with this AIU				
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.				

9.5. CAIU performance counter parameters

TABLE 9-11: CAIU PERFORMANCE COUNTER PARAMETERS

nPerfCounters (CAIU/IOAIU)	Architecture		Release		Visibility: User
					Default
	Min	*Max*	*Min*	*Max*	
Value	0	16	0	8	4
Constraints	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.

TABLE 9-12: CAIU LATENCY COUNTER PARAMETERS

					Visibility: Engg
nLatencyCounters	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	16	16
Constraints	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 CAIU.				
Engineering Description	Parameter applies only to AIUs, DMIs and DIIs only and can be set individually.				

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

nProcessors	**Architecture**	**Release**	**Default**
	Enum	*Enum*	
Value	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
Constraints	Applied to CHI-AIU and IOAIU.		
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.		

TABLE 9-14: AXIDPROCSELECTBITS PARAMETERS FOR CAIU

AxIdProcSelectBits	**Architecture**	**Release**	**Default**
	Integer Array	*Integer Array*	
Value			
Constraints	Applied Only to IOAIU.		
Customer Description	Processor Select Bits from AXID. If there is only one processor, the array is empty and is default as zero		
Engineering Description			

9.7. CAIU SysCmd Hardware parameters

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

TABLE 9-15: USESYSCOENGINE PARAMETERS FOR CAIU

Name: useSysCoEngine			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	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		

TABLE 9-16: USESYSREQSENDER PARAMETERS FOR CAIU

Name: useSysReqSender			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	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		

TABLE 9-17: USESYSREQRECEIVER PARAMETERS FOR CAIU

Name: useSysReqReceiver			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	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		

9.8. CAIU Connectivity parameters

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

TABLE 9-18: HEXAIUDCEVEC PARAMETERS FOR CAIU

Name: hexAiuDceVec					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	FFFFFFFF	0	FFFF	1
Constraints	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				

TABLE 9-19: HEXAIUDMIVEC PARAMETERS FOR CAIU

Name: hexAiuDmiVec					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	*0*	*FFFFFFFF*	*0*	*FFFF*	*1*
Constraints	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				

TABLE 9-20: HEXAIUDIIVEC PARAMETERS FOR CAIU

Name: hexAiuDiiVec					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	FFFFFFFF	1	FFFF	1
Constraints	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				

TABLE 9-21: HEXAIUCONNECTEDDCEFUNITID PARAMETERS FOR CAIU

Name: hexAiuConnectedDceFunitId					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	FFFFFFFF	0	FFFF	1
Constraints	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.				

TABLE 9-22: NAIUCONNECTEDDCES PARAMETERS FOR CAIU

Name: nAiuConnectedDces					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	64	1	16	1
Constraints	Number of DCEs connected to this each AIU.				
Customer Description					
Engineering Description	Specifies the number of caching agents (AIUs) that are connected to DCE				

10. DCE User Settable Parameters

10.1. DCE resource parameters

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.

Table 10-1: nAttCtrlEntries parameter

nAttCtrlEnries	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	4	96	4	96	4
Constraints					
Customer Description	Specify the maximum number of active coherent transactions tracked by each DCE.				
Engineering Description					

Table 10-2: nCMDSkidBufSize parameter

nCMDSkidBufSize	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	4	768	4	768	16
Constraints	restrict granularity, supporting only sizes: nCMDSkidBufArb + {0, 4, 8, 16, 32, 64, 96, 128, 160, 192, 224, 256,512} Maximum value is limited to 768.				
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 maximium value of nCMDSkidBufSize should be the total sum of nDceCmdCredits as defined in Table 9-5 from any connected CAIU(s) and nDceCmdCredits as defined in Table 14-5 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.

[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

Table 10-3: nCMDSkidBufArb parameter

nCMDSkidBufArb	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	4	256	4	256	16
Constraints	≤ 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				

10.2. DCE credit parameters

Table 10-4: nDceRbCredits parameter

nDceRbCredits	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	2	64	2	64	2
Constraints					
Customer Description	Number of RB credits per DCE 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.				
Engineering Description	Number of RB credits per DCE				

Table 10-5: nAiuSnpCredits parameter

nAiuSnpCredits	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	2	16	2	16	2
Constraints					
Customer Description	Specify the maximum number of snoop request credits per AIU.				
Engineering Description					

Table 10-6: nDmiMrdCredits parameter

nDmiMrdCredits	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	2	16	2	16	2
Constraints					
Customer Description	Specify the maximum number of memory read credits per DMI.				

Engineering Description	

10.3. DCE snoop filter parameters

Table 10-7: nSets SF Configuration parameters

nSets	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	16	128 K	16	128 K	16
Constraints	Number of sets must be divisible by number of DCEs in the system. The result of this number divided by the number of DCEs in an Ncore should be a power-of-two number to simplify memory cell selection, and is equal to or less than 32,768.				
Customer Description	Number of sets for the selected snoop filter.				
Engineering Description	Maximum sets are increased to 128 K sets in 3.7 to support high performance configurations.				

Table 10-8: nWays SF Configuration parameters

nWays	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	2	32	2	32	4
Constraints					
Customer Description	Number of ways for the selected snoop filter				
Engineering Description					

Table 10-9: nVictimEntries SF Configuration parameters

nVictimEntries	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	64	0	64	2
Constraints					
Customer Description	Number of victim buffer entries for the specified snoop filter, per DCE				
Engineering Description					

Table 10-10: aPrimaryBits parameters

aPrimaryBits	**Architecture**	**Release**	**Default**
	Array of Integers	*Array of Integers*	
Value			
Constraints	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].		

Customer Description	Set selection parameter.
Engineering Description	

TABLE 10-11: MEMORY PARAMETER FOR DCE SNOOP FILTER

Memory	**Architecture**	**Release**	**Default**
Constraints			
Engineering Description	This parameter is to assign SRAM. For the memory setting, refer Table 21-6		
NOTE: The max number of snoop filter is constrained to be no more than Snoop agents. Snoop filter must have an aiu assigned.			

TABLE 10-12: REMOTECACHINGAGENTS PARAMETER

Name: *RemoteCachingAgents*		**Type: array of strings**	**Visibility:** User
	Architecture	**Release**	**Default**
	array of strings	array of strings	
Value			
Constraint	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)		

TABLE 10-13: LOCALCACHINGAGENTS PARAMETER

Name: *LocalCachingAgents*		**Type: array of strings**	**Visibility:** User
	Architecture	**Release**	**Default**
	array of strings	array of strings	
Value			
Constraint	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)		

10.4. DCE performance counter parameters

TABLE 10-14: DCE NPERFCOUNTERS PARAMETERS

nPerfCounters	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	16	0	16	0
Constraints	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.				

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.

10.5. 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 10-15: NTAGGEDMONITORS PARAMETER

Name: nTaggedMonitors					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	8	0	8	0
Constraints					
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					

10.6. 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 10-16: HEXDCECONNECTEDDMIFUNITID PARAMETER

Name: hexDceConnectedDmiFunitID					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	FFFFFFFF	0	FFFF	1
Constraint	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.				
Customer Description					
Engineering Description	This must be a port in RTL (tACHL) and tie off parameter in SW List of DMI FuntiIDs that are connected to the DCE				

Table 10-17: hexDceConnectedCaFunitID parameter

Name: hexDceConnectedCaFunitID					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	FFFFFFFF	0	FFFF	1
Constraint	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				
Customer Description					
Engineering Description	This must be a port in RTL (tACHL) and tie off parameter in SW List of caching agent FuntiIDs that are connected to the DCE				

Table 10-18: hexDceDmiVec parameter

Name: hexDceDmiVec					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	FFFFFFFF	0	FFFF	1
Constraint	Size of the vector is equal to the number of DMIs in the system				
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 DCE is connected to the associated DMI at that NunitID				

Table 10-19: hexDceDmiRbOffset parameter

Name: hexDceDmiRbOffset					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	Max 32 DMIs	0	Max 16 DMIs	1
Constraint	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.				

Table 10-20: nDceConnectedCas parameter

Name: nDceConnectedCas					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	64	1	32	1
Constraint	Number of Caching agents connected to each DCE. This parameter must be same for all DCEs				
Customer Description					
Engineering Description	Specifies the number of caching agents (AIUs) that are connected to DCE				

Table 10-21: nDceConnectedDmis parameter

Name: nDceConnectedDmis					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	32	1	16	1
Constraint	Number of DMIs connected to each DCE. This parameter must be same for all DCEs				
Customer Description					
Engineering Description	Specifies the number of DMIs that are connected to DCE				

Table 10-22: nDceRbCredits parameter

Name: nDceRbCredits					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	2	64	2	64	2
Constraint	Number of RB credits per DCE The value is same for all DCEs and DMIs				
Customer Description					
Engineering Description	Number of RB credits per DCE				

11. *DMI User Settable Parameters*

11.1. DMI resource parameters

TABLE 11-1: GENERICPORTS PARAMETERS FOR DMI

genericports	Architecture	Release	Default
Constraints			
Engineering Description	To assign user defined ports for place holder definition (Resiliency); Described in Chapter 21.1		

TABLE 11-2: NRTTCTRLENTRY PARAMETERS

nRttCtrlEnries	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	4	128	4	128	4
Constraints					
Customer Description	Specify the number of allowed outstanding read transactions on the downstream AXI interface.				
Engineering Description					

TABLE 11-3: NWTTCTRLENTRY PARAMETERS

nWttCtrlEnries	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	4	64	4	64	4
Constraints					
Customer Description	Specify number of allowed outstanding write transactions on the downstream AXI interface.				
Engineering Description					

TABLE 11-4: NDMIRBCREDITS PARAMETER

nDmiRbCredits	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	2	64	2	64	2
Constraints					
Customer Description	Specify the maximum number of non-coherent write request buffer credits.				
Engineering Description					

TABLE 11-5: NCMDSKIDBUFSIZE PARAMETER

nCMDSkidBufSize	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	4	768	4	768	16
Constraints	Restrict granularity, supporting only sizes: nCMDSkidBufArb + {0, 4, 8, 16, 32, 64, 96, 128, 160, 192, 224, 256,512} Maximum value is limited to 768.				
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 9-6 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))[5], which assuming the minimum nDmiCmdCredits of 2 from each connected agent.

TABLE 11-6: NCMDSKIDBUFARB PARAMETER

nCMDSkidBufArb	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	4	256	4	256	16
Constraints	≤ 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				

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

TABLE 11-7: NMRDSKIDBUFSIZE PARAMETER

nMrdSkidBufSize	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	4	768	4	768	16
Constraints	Restrict granularity, supporting only sizes: nMrdSkidBufArb + {0, 4, 8, 16, 32, 64, 96, 128, 160, 192, 224, 256, 512} Maximum value is limited to 768.				
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 as defined in Table 10-6 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))[5], which assuming the minimum nDmiMrdCredits of 2 from each connected DCE.

TABLE 11-8: NMRDSKIDBUFARB PARAMETER

nMrdSkidBufArb	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	4	256	4	256	16
Constraints	≤ nMrdSkidBufArb restricted granularity, support only sizes: 4, 8, 16, 32, 48, 64, 128, 192, 256 Maximum value is limited to 320.				
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				

Table 11-9: nUseMemRspIntrlv parameter

nUseMemRspIntrlv	Architecture		Release		Default
Value	True	False	True	False	False
Constraints					
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.				

Table 11-10: nExclusiveEntries Parameter

nExclusiveEntries	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	0	8	0	8	0
Constraints					
Customer Description	defines the number of exclusive monitors				
Engineering Description	A value of 0 means no exclusive monitor will be instantiated				

11.2. DMI address map parameters

Table 11-11: nAddrTransRegisters Parameters

nAddrTransRegisters	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	0	4	0	4	0
Constraints					
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					

11.3. DMI System Cache parameters

Table 11-12: DMI System Cache Enable Parameters

Name: hasSysMemCache			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid values*	
Value	True, False	True, False	False
Constraints			
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			

Table 11-13: DMI Scratchpad Enable Parameters

useScratchPad	**Architecture**	**Release**	**Default**
	Valid Values	*Valid values*	
Value	True, False	True, False	False
Constraints	Can be enabled only when system cache is enabled.		
Customer Description	Enable ScratchPad		
Engineering Description			

Table 11-14: DMI Cache Way Partitioning Registers parameters

nWayPartitioningRegisters	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	16	0	16	0
Constraints					
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					

Table 11-15: DMI Cache nTagBank Configuration parameters

nTagBanks	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	4	1	4	1
Constraints	Values limited to 1, 2, 4				
Customer Description	Number of Tag banks.				
Engineering Description					

TABLE 11-16: DMI CACHE NDATABANK CONFIGURATION PARAMETERS

nDataBanks	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	1	4	1	4	1
Constraints	Values limited to 1, 2, 4				
Customer Description	Number of Data banks.				
Engineering Description					

TABLE 11-17: MEMORY PARAMETER SMC

Memory	Architecture	Release	Default
Constraints			
Engineering Description	This parameter is to assign SRAM. For the memory setting, refer Table 21-8 and Table 21-9		

11.4. DMI performance counter parameters

TABLE 11-18: DMI NPERFCOUNTERS PARAMETERS

nPerfCounters	Architecture		Release		Visibility: User Default
	Min	*Max*	*Min*	*Max*	
Value	0	16	0	16	4
Constraints	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.				

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 11-19: DMI LATENCY COUNTER PARAMETERS

nLatencyCounters	Architecture		Release		Visibility: Engg Default
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	16	16
Constraints	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 individually.				

11.5. 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/E, AXI5 and ACE5-LITE interface once the corresponding Atomic_Transactions property is enabled. Thus, in Ncore 3 FAOs are supported for all locations in system memory connected via the DMI.

TABLE 11-20: USEATOMIC PARAMETER

useAtomic	Architecture	Release	**Visibility: User** Default
	Data Type: Boolean	*Data Type: Boolean*	
Value	True/False	True/False	False
Constraints	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.		

11.6. DMI QoS Enhancement parameters

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

1. 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”
2. 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 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.

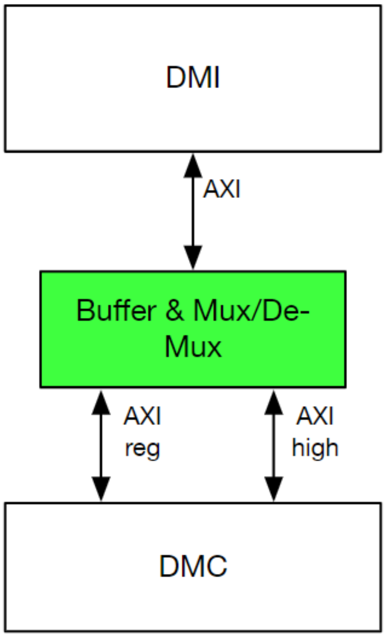


TABLE 11-21: DMIQOSTHVAL PARAMETERS

Name: DmiQoSThVal					Visibitity: User
	Architecture		**Release**		**Default**
	Min	*Max*	*min*	*max*	
Value	1	15	1	15	8
Constraints	This parameter is available only when QoS(Table 4-19) 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					

TABLE 11-22: NDMIWTTQOSRSV PARAMETERS

Name: nDmiWttQoSRsv					Visibitity: User
	Architecture		**Release**		**Default**
	Min	*Max*	*min*	*max*	
Value	1	64	1	32	1
Constraints	This parameter is available only when QoS is enabled. 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
Customer Description	WTT entries in DMI reserved for high priority hard real time traffic.
Engineering Description	

Table 11-23: nDmiRttQoSRsv Parameters

Name: nDmiWttQoSRsv					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*min*	*max*	
Value	1	64	1	32	1
Constraints	This parameter is available only when QoS is enabled. Maximum acceptable value must be RTT size - 1				
Customer Description	RTT entries in DMI reserved for high priority hard real time traffic.				
Engineering Description					

11.7. DMI addressBits parameter for custom AxiId address bit selection

Table 11-24: addressBits Parameter

Name: addressBits		**Type: Array**	**Visibility: user Settable**
	Architecture	Release	Default
	Integer Array	Integer Array	
Value			
Constraint	Applied only to DII.		
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: wAwid Default Array Size: 0 (no entries) Array entry integer range: 0 to (system.concertocparams.wAddr-1) addressBits = addressIdMap.addressBits Only applicable for Writes:		

	A similar configurability as DII will be used to generate the ID of write transactions. By default, the axid is generated from the LSB of the address after the offset has been striped i.e bits 6 to 6+wAxid-1. Maestro will now allow user to specify the bits to use from the address. It will be passed as an array of integer; each integer must be greater or equal to 6 and smaller than wAddr-1. The number of elements in the array will be between 0 (empty) and wAxid. The ID will be generated from the index specified in the array, starting from Javascript index 0 to generate the LSB of the Axid. If the numer of element is smaller than the LSB, the remaining bits that will be used will be the same as before i.e the natural location. Example: - array=[6,8,10], wAxid=5: Axid will be : Addr[10,9,6,8,10] (using Verilog convention of lsb on the right) - array=[], wAxid=2 : Axid will be : Addr[7,6] - array=[15,14,13,12], wAxid = 4 :Axid will be : Addr[12,13,14,15] (using Verilog convention of lsb on the right)

12. *DII User Settable Parameters*

12.1. DII resource parameters

Table 12-1: nRttCtrlEntry Parameters

Name: nRttCtrlEnries					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*min*	*max*	
Value	4	32	4	32	4
Constraints					
Customer Description	Specify number of outstanding read transactions on the AXI interface.				
Engineering Description					

Table 12-2: nWttCtrlEntry Parameters

Name: nWttCtrlEnries					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*min*	*max*	
Value	4	32	4	32	4
Constraints					
Customer Description	Specify number of outstanding write transactions on the AXI interface.				
Engineering Description					

Specify the size of the largest endpoint device connected to the DII. The size is in KBs. This size isued to achive endpoint ordering as defined by ARM CHI specification

Table 12-3: nLargestEndPoint parameter

Name: nLargestEndpoint					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*min*	*max*	
Value	4	2^39	4	2^39	4
Constraints					
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					

Note: Why are we allowing such a large value - could 2^{32} be sufficient?

A: depending on the downstream size, can connect with an FlexNoC and create a large endpoint space

Table 12-4: nDiiRbCredits Parameter

Name: nDiiRbCredits					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*min*	*max*	
Value	2	32	2	32	2
Constraints					
Customer Description	Specify the maximum number of non-coherent write request buffer credits.				
Engineering Description					

TABLE 12-5: NCMDSKIDBUFSIZE PARAMETER

nCMDSkidBufSize	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	4	768	4	768	16
Constraints	Restrict granularity, supporting only sizes: nCMDSkidBufArb + {0, 4, 8, 16, 32, 64, 96, 128, 160, 192, 224, 256, 512} Maximum value is limited to 768.				
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 9-7 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))[6], which assuming the minimum nDiiCmdCredits of 2 from each connected agent.

TABLE 12-6: NCMDSKIDBUFARB PARAMETER

nCMDSkidBufArb	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	4	256	4	256	16
Constraints	≤ 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				

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

TABLE 12-7: NEXCLUSIVEENTRIES PARAMETER

nExclusiveEntries	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	0	8	0	8	0
Constraints					
Customer Description	defines the number of exclusive monitors				
Engineering Description	A value of 0 means no exclusive monitor will be instantiated				

12.2. DII address map parameters

TABLE 12-8: NADDRTRANSREGISTERS PARAMETER FOR DII

Name: nLargestEndpoint					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*min*	*max*	
Value	0	16	0	8	4
Constraints					
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				

12.3. DII performance monitor parameters

TABLE 12-9: DII NPERFCOUNTERS PARAMETERS

Name: nPerfCounters	Architecture		Release		Visibility: User
					Default
	Min	*Max*	*min*	*max*	
Value	0	16	0	16	0
Constraints	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.				

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 12-10: DII LATENCY COUNTER PARAMETERS

nLatencyCounters	Architecture		Release		Visibility: Engg
					Default
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	16	16
Constraints	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.				

Table 12-11: addressBits Parameter

Name: addressBits		**Type:** Array	**Visibility:** user Settable
	Architecture	**Release**	**Default**
	Integer Array	*Integer Array*	
Value			
Constraint	Applied only to DII.		
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 three. For example: a. Awid[x] = addressBits[wCachelineOffset+3] b. Awid[x+1] = addressBits[wCachelineOffset+1+3] …		

13. *DVE User Settable Parameters*

13.1. DVE resource parameters

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

TABLE 13-1: MEMORY PARAMTER FOR DVE

Memory	Architecture	Release	Default
Constraints			
Engineering Description	This parameter is to assign SRAM. For the memory setting, refer Table 21-10		

TABLE 13-2: DVE EVENTBROADCASTERFIFODEPTH PARAMETER

Name: EventBroadcasterFIFOdepth			Visiblity: Engg
	Architecture	**Release**	**Default**
	integer	*integer*	
Value	Sum(useSysReqSender)	Sum(useSysReqSender)	Sum(useSysReqSender)
Constraints	Add up total number of useSysReqSender of every units		
Customer Description			
Engineering Description	Used to size FIFO of Event Broadcaster hardware in the DVE		

13.2. DVE performance monitor parameters

TABLE 13-3: DVE NPERFCOUNTERS PARAMETERS

nPerfCounters	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	0	16	0	16	4
Constraints	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.				

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.

13.3. System level credit 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

Table 13-4: nDvmCmdCredits Parameter

Name: nDvmCmdCredits			Type: int		Visibiltiy: User
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	2	4	2	4	2
Constraints	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.				

Table 13-5: nDvmSnpCredits Parameter

Name: nDvmSnpCredits		Type: Int	Visibility: **Engg**
	Architecture	**Release**	**Default**
	Enum	*Enum*	
Value	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
Constraints	Must be a multiple of 2		
Customer Description	If noDVM[7] 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		

[7] It is the noDVM value defined in Table 4-30, which globally turns off the DVM functionalities throughout an Ncore

TABLE 13-6: DVMVERSIONSUPPORT PARAMETER

Name: *DVMVersionSupport*		**Type:** Int	**Visibility:** User
	Architecture	**Release**	**Default**
	Int	*Int*	
Value	{8,0}, {8,1}, {8,4}	{8,0}, {8,1}, {8,4}	{8,4}
Format	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}		
Constraint	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		

14. NCAIU User Settable Parameters

14.1. NCAIU multiport parameters

TABLE 14-1: NNATIVEINTERFACEPORTS PARAMETER FOR NCAIU

nNativeInterfacePorts	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	8	1	8	1
Constraints	Valid values are power of 2s. 1, 2, 4, or 8.				
Customer Description	Specifies the number of native interface ports				
Engineering Description					

TABLE 14-2: ANCAIUINTVFUNC PARAMETER FOR NCAIU

aNcaiuIntvFunc	**Architecture**	**Release**	**Default**
Parameters	*Name*	*Name*	
	aPrimaryBits	*aPrimaryBits*	Array of integers
	aSecondaryBits	Not user visible	Array of strings
Constraints	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			

14.2. NCAIU resource parameters

TABLE 14-3: NOTTCTRLENTRIES FOR NCAIU

nOttCtrlEntries	**Architecture**	**Release**	**Default**

	Min	*Max*	*Min*	*Max*	
Value	4	1024	8	1024	32
Constraints	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					

TABLE 14-4: MEMORY PARAMETER FOR NCAIU

Memory	**Architecture**		**Release**		**Default**
Constraints					
Engineering Description	For the memory setting, refer Table 21-5 and Table 21-9				

14.3. NCAIU credit parameters

TABLE 14-5: NDCECMDCREDITS FOR NCAIU

nDceCmdCredits	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	2	32	2	32	32
Constraints	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					

TABLE 14-6: NDMICMDCREDITS FOR NCAIU

nDmiCmdCredits	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	2	32	2	32	16
Constraints	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					

TABLE 14-7: NDIICMDCREDITS FOR NCAIU

nDiiCmdCredits	**Architecture**		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	2	32	2	32	16
Constraints	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	

14.4. NCAIU address map parameter

TABLE 14-8: FNCSRACCESS_PARAMETER

fnCsrAccess	Architecture	Release	Default
	Valid Values	*Valid values*	
Value	True, False	True, False	false
Constraints	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			

14.5. NCAIU snoop filter parameters

TABLE 14-9: SNOOPFILTERASSIGNMENT PARAMETER

snoopFilterAssignment	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	0	64	0	64	0
Constraints	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.				

14.6. NCAIU proxy cache parameters

TABLE 14-10: PROXY CACHE ENABLE PARAMETERS

Name: hasProxyCache			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid values*	
Value	True, False	True, False	False
Constraints	This option enables a ProxyCache is configured for a NCAIU.		
Customer Description	This option adds a ProxyCache in NCAIU.		
Engineering Description			

Table 14-11: Proxy Cache nTagBank Configuration parameters

nTagBanks	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	1	2	1	2	1
Constraints	Values limited to 1, 2				
Customer Description	Number of Tag banks.				
Engineering Description					

Table 14-12: Proxy Cache nDataBank Configuration parameters

nDataBanks	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	1	4	1	4	1
Constraints	Values limited to 1, 2, 4				
Customer Description	Number of Data banks.				
Engineering Description					

Table 14-13: Proxy Cache nSets Configuration parameters

nSets	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	16	8192	2	8192	16
Constraints	Number of sets must be power of 2 number				
Customer Description	Number of sets for proxy cache				
Engineering Description					

Table 14-14: Proxy Cache nWays Configuration parameters

nWays	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	2	16	2	16	2
Constraints	Available number of ways for a proxy cache are: 2, 4, 8, and 16				
Customer Description	Number of ways for proxy cache				
Engineering Description					

14.7. NCAIU performance counter parameters

TABLE 14-15: NPERFCOUNTERS PARMAETER FOR NCAIU

nPerfCounters	Architecture		Release		**Visibility:** Engg Default
	Min	*Max*	*Min*	*Max*	
Value	0	16	0	8	0
Constraints	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.				

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 14-16: NLATENCYCOUNTERS PARMAETER FOR NCAIU

nLatencyCounters	Architecture		Release		**Visibility:** Engg Default
	Min	*Max*	*Min*	*Max*	
Value	0	32	0	16	16
Constraints	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.				

14.8. NCAIU disable read data interleaving parameters

TABLE 14-17: FNDISABLERDINTERLEAVE PARMAETER FOR NCAIU

fnDisableRdInterleave	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	0	1	0	1	0
Constraints					
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				

14.9. NCAIU SysCmd Hardware parameters

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.

TABLE 14-18: USESYSCOENGINE PARAMETERS FOR NCAIU

Name: useSysCoEngine			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	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		

TABLE 14-19: USESYSREQSENDER PARAMETERS FOR NCAIU

Name: useSysReqSender			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	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		

TABLE 14-20: USESYSREQRECEIVER PARAMETERS FOR NCAIU

Name: useSysReqReceiver			Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	*True, False*	*True, False*	True
Constraints	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		

14.10. NCAIU Connectivity parameters

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

TABLE 14-21: HEXAIUDCEVEC PARAMETERS FOR NCAIU

Name: hexAiuDceVec					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	FFFFFFFF	0	FFFFFFF	1
Constraints	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				

TABLE 14-22: HEXAIUDMIVEC PARAMETERS FOR NCAIU

Name: hexAiuDmiVec					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	FFFFFFFF	0	FFFF	1
Constraints	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				

TABLE 14-23: HEXAIUDIIVEC PARAMETERS FOR NCAIU

Name: hexAiuDiiVec					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	FFFFFFFF	0	FFFF	1
Constraints	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				

TABLE 14-24: HEXAIUCONNECTEDDCEFUNITID PARAMETERS FOR NCAIU

Name: hexAiuConnectedDceFunitId					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	0	FFFFFFFF	0	FFFF	1
Constraints	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.				

TABLE 14-25: NAIUCONNECTEDDCES PARAMETERS FOR NCAIU

Name: nAiuConnectedDces					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	1	64	1	32	1
Constraints	Number of DCEs connected to this each AIU.				
Customer Description					
Engineering Description	Specifies the number of caching agents (AIUs) that are connected to DCE				

14.11. PCAIU OWO parameters

TABLE 14-26: ORDEREDWRITEOBSERVATION PARAMETER

Name: orderedWriteObservation		Type: *Boolean*	Visibility: User
	Architecture	**Release**	**Default**
Value	True/False	True/False	False
Constraints	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.		
Customer Description	The Ordered Write Observation feature to accelerate PCIe traffic		
Engineering Description	The Ordered Write Observation feature to accelerate PCIe traffic		

TABLE 14-27: MULTICYCLEODSRAM PARAMETER

Name: multicycleODSRAM		Type: *Boolean*	Visibility: User
	Architecture	**Release**	**Default**
Value	True/False	True/False	False
Constraints	This parameter only takes effect and be visible if orderedWriteObservation parameter is set to TRUE. It is mainly used to ease timing closure for a PCAIU configuration once OWO is enabled and the Outstanding Data (OD) Buffer is big.		
Customer Description	OD timing closure optimization parameter for OWO configuration		
Engineering Description	OD timing closure optimization parameter for OWO configuration		

TABLE 14-28: PCAIU DATAWIDTH PARAMETER

Name: IOAIUpDataWidth		Type: **Enum**	Visibility: User
	Architecture	**Release**	**Default**
Value	{256, 512}	{256, 512}	256
Constraints	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 PCAIU 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 PCAIU 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		

Once IOAIUpDataWidth parameter for a PCAIU is set to 512, whatever value taken in Ncore system level syncDepth parameter defined in Table 4-26 also takes effect to all the synchronizers employed inside the asynchronous frequency adaption gasket inserted automatically by Maestro.

TABLE 14-29: NENTRIESINSHIM PARAMETER

Name: nEntriesInShim			Type: *Integer*		Visibility: Engg
	Architecture		**Release**		**Default**
Value	Min	Max	Min	Max	32
	4	1024	4	1024	
Constraints	This parameter only takes effect and be visible if orderedWriteObservation parameter is set to TRUE **AND** IOAIUpDataWidth is set to 512. 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 a PCAIU can support, which should be the same value as what it is defined for the nOttCtrlEntries defined in Table 14-3. And it will be derived internally once a 512 bit interface is used.				

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 a PCAIU can support.
Engineering 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 a PCAIU can support.

15. *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 15-1: nSets parameters of CCP

Name: nSets			Visibitity: User
	Architecture	**Release**	**Default**
	Valid Values	*Boolean*	
Value	16', '32', '64', '128', '256', '512', '1024', '2048', '4096', '8192	16', '32', '64', '128', '256', '512', '1024', '2048', '4096', '8192	16
Constraints	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. Expect log2(nSets) bits for primary selection bits. Must be multiple of nNativeInterfacePorts for both min and max ranges and actual value.		
Customer Description	Specify the number of sets/entries in the Cache.		
Engineering Description			

Table 15-2: nWays parameter of CCP

Name: nWays					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*	*min*	*max*	
Value	2	16	2	16	2
Constraints					
Customer Description	Specify the number of sets/entries in the Cache.				
Engineering Description					

Table 15-3: useScratchPad parameter of CCP

Name: useScratchPad			Visibility: User
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	FALSE
Constraints			
Customer Description	Enable Scratchpad. The visibility will be overridden based on block type.		
Engineering Description			

Table 15-4: PriSubDiagAddrBits parameters

Name: PriSubDiagAddrBits			Visibility: User
	Architecture	**Release**	**Default**
	Array of strings	*Array of strings*	
Value			
Constraints	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		
Customer Description	Specify address bits to be used as primary set select bits.		
Engineering Description			

Table 15-5: TagBankSelBits parameters

Name: TagBankSelBits			Visibility: User
	Architecture	**Release**	**Default**
	Array of strings	*Array of strings*	
Value			
Constraints			
Customer Description	The tag bank select 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).		

Engineering Description	

TABLE 15-6: DATABANKSELBITS PARAMETERS

Name: DataBankSelBits			Visibility: User
	Architecture	**Release**	**Default**
	Array of strings	*Array of strings*	
Value			
Constraints	The data bank select 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.		
Customer Description	Specify data bank select bit. This bit must be one of the bits from the primary select bits.		
Engineering Description			

TABLE 15-7: CACHEREPLPOLICY PARAMETER

Name: cacheReplPolicy		**Type:** Enum	**Visibility:** user Settable
	Architecture	**Release**	**Default**
	Enum	*Enum*	
Value	RANDOM, NRU, SRRIP, pLRU	RANDOM, NRU, pLRU	RANDOM
Constraint	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		

TABLE 15-8: CACHEREPLSTATEWIDTH PARAMETER

Name: *cacheReplStateWidth*		**Type:** Int	**Visibility:** derived
	Architecture	**Release**	**Default**
	Int	*Int*	
Value	0, 1, 2	0, 1	0
Constraint	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.		

16. *Tiling Related Parameters*

TABLE 16-1: PORT_ID PARAMETER FOR PACKETIZER

Name: Port_ID		**Type Binary**	Visibility: Engg
	Architecture	**Release**	**Default**
	Binary	*Binary*	
Value	{1'b0, 1'b1}	{1'b0, 1'b1}	1'b0
Constraints	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.		

TABLE 16-2: ROUTEONSID PARAMETER FOR PACKETIZER

Name: routeOnSid		**Type Boolean**	Visibility: Engg
	Architecture	**Release**	**Default**
	Boolean	*Boolean*	
Value	True, False	True, False	False
Constraints	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	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.		
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.		

17. *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

17.1. sym_switch/sym_buf_switch

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

TABLE 17-1: SYM_BUF_SWITCH AND SYM_SWITCH PARAMETER: PORTDATAWIDTH

Name: portDataWidth			Visibility: User
	Architecture	**Release**	**Default**
	Valid values	*Valid Values*	
Value	64, 128, 256	64, 128, 256	256
Constraints			
Customer Description	Data width of all ports of the switch		
Engineering Description	Applied to sym_switch and sym_buf_switch		

TABLE 17-2: SYM_BUF_SWITCH PARAMETER: INPUTBUFFERDEPTH

Name: inputBufferDepth			Visibility: User
	Architecture	**Release**	**Default**
Value	[0, 2, 4, 8, 12, 16, 24, 32]	[0, 2, 4, 8, 12, 16, 24, 32]	2
Constraints			
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.		

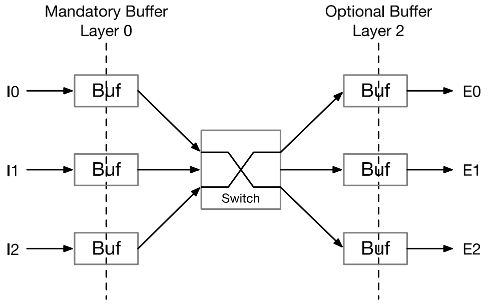


FIGURE 17-1: SYM_BUF_SWITCH IN CDTI, WITH ONLY ONE VC

17.2. 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

17.3. 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.

17.4. sym_nRate_adapter

Please refer to section 20.5 for more details.

17.5. 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.

17.6. sym_pipe_adapter

TABLE 17-3: SYM_PIPE_ADAPTER PARAMETER: DEPTH

Name: depth			Visibility: User
	Architecture	**Release**	**Default**
Value	[0,1,2,3]	[1,2]	2
Constraints			
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.		

18. *Derived/Fixed Socket Parameters*

18.1. AXI Interface

TABLE 18-1: AXI INTERFACE FIXED PARAMETERS

Parameter Name	Type		Default	Min	Max	Enum	Description
wResp	Integer	Fixed	2	2	4		
wWUser	Integer	Fixed	0	0	16	Not being used	
wBUser	Integer	Fixed	0	0	200	Not being used	
wRUser	Integer	Fixed	0	0	200	Not being used	
wLen	Integer	Fixed	8	8	8		
wSize	Integer	Fixed	3	3	4	Only 3. Because we are not supporting 512.	
wLock	Integer	Fixed	1	0	1	Always 1	Exclusive accesses are always supported by default
wQos	Integer	Fixed	0	0	4	['0', '4']	
wRegion	Integer	Fixed	0	0	4	Fixed as 0	
wProt	Integer	Fixed	3			Fixed as 3	

18.2. APB Interface

TABLE 18-2: APB INTERFACE FIXED PARAMETERS

Parameter Name	Type		Default	Min	Max	Enum	Description
wAddr	Integer	Fixed	12	12	64		
wData	Integer	Fixed	32	32	32		
wProt	Integer	Fixed	0	0	3	['0', '3']	3 if APB4
wStrb	Integer	Fixed	0	0	4	['0', '1', '2', '4']	4 if APB4
wPSlverr	Integer	Fixed	0	0	1		

18.3. ACE Interface

TABLE 18-3: ACE INTERFACE USER SETTABLE PARAMETERS

Parameter Name	Type		Default	Min	Max	Description/Derivation
eUnique	Integer	Eng. Param.	1	0	1	
wCdData	Integer	Fixed	0			
wSnoop	Integer	Eng. Param.	3	3	3	
eAc	Integer	Fixed	1	1	1	
wResp	Integer	Fixed	4	2	4	
eDomain	Integer	Fixed	1	1	1	Drives shareableTransactions property and it is always set as TRUE
useQos	Boolean	Derived				useQoS: system parameter
wQos	Integer	Derived				wQos = (useQos) ? 4 : 0;

18.4. ACE5-LITE Interface

TABLE 18-4: ACE5-LITE INTERFACE DERIVED PARAMETERS

Parameter Name	Type		Default	Min	Max	Description/Derivation
wLoop	Integer	Fixed	0	0	0	
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	0	0	
wCdData	Integer	Fixed	0			
wSnoop	Integer	Derived	3	3	4	wSnoop = eStash == 1 ? 4 : 3;
eStash	Integer	Fixed	1	1	1	
eAtomic	Integer	Fixed	1	1	1	
eDomain	Integer	Fixed	1	1	1	Drives shareableTransactions property and it is always set as TRUE

18.5. ACE-LITE Interface

TABLE 18-5: ACE-LITE INTERFACE DERIVED PARAMETERS

Parameter Name	Type		Default	Min	Max	Description/Derivation
wLoop	wLoop	Fixed	0	0	0	
eTrace	eTrace	Fixed	0	0	0	
eUnique	eUnique	Fixed	0	0	0	
wCdData	wCdData	Fixed	0			
wSnoop	wSnoop	Fixed	3	3	3	
eStash	eStash	Fixed	0	0	0	
eAtomic	eAtomic	Fixed	0	0	0	
eDomain	eDomain	Fixed	1	1	1	Drives shareableTransactions property and it is always set as TRUE

18.6. CHI_B Interface

TABLE 18-6: 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	8	
ReturnNID	Integer	Derived	7	7/11	ReturnNID = NodeID_Width;
StashNIDValid	Integer	Fixed	1	1	
ReturnTxnID	Integer	Fixed	8	8	
REQ_Opcode	Integer	Fixed	6	6	
RSP_Opcode	Integer	Fixed	4	4	
SNP_Opcode	Integer	Fixed	5	5	
DAT_Opcode	Integer	Fixed	3	3	
Size	Integer	Fixed	3	3	
wAddr	Integer	Fixed	48	44/52	Physical address width wAddr = {44, 48, 52}
NS	Integer	Fixed	1	1	
LikelyShared	Integer	Fixed	1	1	
AllowRetry	Integer	Fixed	1	1	
Order	Integer	Fixed	2	2	
PCrdType	Integer	Fixed	4	4	
MemAttr	Integer	Fixed	4	4	
SnpAttr	Integer	Fixed	1	1	
LPID	Integer	Fixed	5	5	
Excl	Integer	Fixed	1	1	
ExCompAck	Integer	Fixed	1	1	
TraceTag	Integer	Fixed	1	1	
DAT_RSVDC	Integer	Fixed	0	0	Not supported and is always fixed at zero
RespErr	Integer	Fixed	2	2	
Resp	Integer	Fixed	3	3	
FwdState	Integer	Fixed	3	3	
DBID	Integer	Fixed	8	8	
FwdNID	Integer	Derived	7	7/11	FwdNID = NodeID_Width;
FwdTxnID	Integer	Fixed	8	8	
DoNotGoToSD	Integer	Fixed	1	1	

Table 18-7: CHI_B Interface Derived Parameters-2

Parameter Name	Type		Default	Min/ Max	Description/Derivation
RetToSrc	Integer	Fixed	1	1	
Homenode_ID	Integer	Derived	7	7	Homenode_ID = NodeID_Width;
CCID	Integer	Fixed	2	2	
DataID	Integer	Fixed	2	2	

BE	Integer	Derived	8	64	BE = wData/8;
					wData = { 64, 128, 256}

wQos	Integer	Fixed	4	4	
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;

18.7. CHI_E Interface

Table 18-8: 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	12	
ReturnNID	Integer	Derived	7	7/11	ReturnNID = NodeID_Width;
StashNIDValid	Integer	Fixed	1	1	

Parameter Name	Type		Default	Min/ Max	Description/Derivation
ReturnTxnID	Integer	Fixed	12	12	
REQ_Opcode	Integer	Fixed	7	7	
RSP_Opcode	Integer	Fixed	5	5	
SNP_Opcode	Integer	Fixed	5	5	
DAT_Opcode	Integer	Fixed	4	4	
Size	Integer	Fixed	3	3	
wAddr	Integer	Fixed	48	44/52	Physical address width wAddr = {44, 48, 52}
NS	Integer	Fixed	1	1	
LikelyShared	Integer	Fixed	1	1	
AllowRetry	Integer	Fixed	1	1	
Order	Integer	Fixed	2	2	
PCrdType	Integer	Fixed	4	4	
MemAttr	Integer	Fixed	4	4	
SnpAttr	Integer	Fixed	1	1	
GroupExtID	Integer	Fixed	3	3	
LPID	Integer	Fixed	5	5	
Excl	Integer	Fixed	1	1	
ExCompAck	Integer	Fixed	1	1	
TraceTag	Integer	Fixed	1	1	
DAT_RSVDC	Integer	Fixed	0	0	Not supported and is always fixed at zero
RespErr	Integer	Fixed	2	2	
Resp	Integer	Fixed	3	3	
FwdState	Integer	Fixed	3	3	
DBID	Integer	Fixed	12	12	
FwdNID	Integer	Derived	7	7/11	FwdNID = NodeID_Width;
FwdTxnID	Integer	Fixed	12	12	
DoNotGoToSD	Integer	Fixed	1	1	

TABLE 18-9: CHI_E INTERFACE DERIVED PARAMETERS-2

Parameter Name	Type		Default	Min/ Max	Description/Derivation
RetToSrc	Integer	Fixed	1	1	
Homenode_ID	Integer	Derived	7	7/11	Homenode_ID = NodeID_Width;
CCID	Integer	Fixed	2	2	
DataID	Integer	Fixed	2	2	

BE	Integer	Derived	8	8/64	BE = wData/8;
					wData = { 64, 128, 256}

wQos	Integer	Fixed	4	4	
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[8]	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;

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

19. *Derived/Fixed Concerto Parameters*

19.1. ConcertoC SMI Param

TABLE 19-1: CONCERTOCSMIPARAM 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, wAXIFIdSet, 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	
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;

TABLE 19-2: CONCERTOCSMIPARAM PARAMETERS

Parameter Name	Type		Default	Min/ Max	Description/Derivation
wSmiSteer	Integer	Derived			wSmiSteer = wSteering;
wSmiTier	Integer	Derived			wSmiTier = wTTier;
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	
wSmiClass	Integer	Fixed	0	0	
wSmiSeqnum	Integer	Fixed	0	0	
wSmiAddr	Integer	Fixed	0	0	
wSmiLen	Integer	Fixed	0	0	
wSmiVNid	Integer	Fixed	0	0	
wSmiProt	Integer	Fixed	0	0	
wSmiTxnHdr	Integer	Fixed	0	0	
nSmiDPvc	Integer	Fixed	1	1	
wSmiDPlast	Integer	Fixed	1	1	
wSmiDPdata	Integer	Derived		128 or 256	Will be defined at block level wSmiDPdata: ncore3 uses 256 max. 512 is not verified
wSmiDPuser	Integer	Fixed	0	0	
wSmiDPbe	Integer	Fixed	0	0	
wSmiDPid	Integer	Fixed	0	0	
wSmiDPerr	Integer	Fixed	0	0	
wSmiDPresp	Integer	Fixed	0	0	
wSmiDPdummy	Integer	Fixed	0	0	

19.2. ConcertoC Param

TABLE 19-3: CONCERTOCPARAM PARAMETERS

Parameter Name	Type	Origin	Default	Min/ Max	Description/Derivation
wCacheLine	Integer	Fixed			Cache line width in byte 64B Cacye Line
wDWId	Integer	Fixed			Number of Bits Identifing a DW Within a CG
wDBad	Integer	Fixed	1		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		
wValid	Integer	Fixed	1		
wReady	Integer	Fixed	1		
wLast	Integer	Fixed	1		
wStashFUnitId	Integer				wStashFUnitId = wFUnitId;
wStashNId	Integer				wStashNId = wStashFUnitId;
HProtEnable	Boolean	Fixed	False		HProt is being defined…?
TTierEnable	Boolean	Fixed	False		Not used in Ncore 3.x
QLEnable	Boolean	Fixed	False		Not used in Ncore 3.x
SteeringEnable	Boolean	Fixed	False		Not used in Ncore 3.x
PriorityEnable	Boolean	Fixed	True		
wTargetId	Integer	Derived			wTargetId = wFUnitId + wFPortId; (from Common)
wInitiatorId	Integer	Derived			wInitiatorId = wFUnitId + wFPortId; (from Common)
wCMType	Integer	Fixed	8	8/8	
wMessageId	Integer	Derived			wMessageId = max({log2MaxAiuCredits, log2MaxDceCredits, log2MaxDmiCredits, log2MaxDiiCredits});

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	0/4	
wSteering	Integer	Fixed	0	0/4	

TABLE 19-4: CONCERTOCPARAM PARAMETERS

Parameter Name	Type	Origin	Default	Min/ Max	Description/Derivation
wPriority	Integer	Derived		0/4	wPriority = useQos ? 3 : 0;
wQL	Integer	Fixed	0	0/4	
wCMHeader	Integer	Derived			wCMHeader = wTargetId + wInitiatorId + wCMType + wMessageId + wHProt + wTTier + wSteering + wPriority + wQL;
wCMStatus	Integer	Fixed	8	8	
wVZ	Integer	Fixed	1	1	
wCA	Integer	Fixed	1	1	
wAC	Integer	Fixed	1	1	
wCH	Integer	Fixed	1	1	
wST	Integer	Fixed	1	1	
wEN	Integer	Fixed	1	1	
wES	Integer	Fixed	1	1	
wNS	Integer	Fixed	1	1	
wPR	Integer	Fixed	1	1	
wOR	Integer	Fixed	2	2	
wLK	Integer	Fixed	2	2	
wRL	Integer	Fixed	2	2	
wTM	Integer	Fixed	1	1	
wUP	Integer	Fixed	2	2	
wPrimary	Integer	Fixed	1	1	
wMW	Integer	Fixed	1	1	
wEO	Integer	Fixed	0		
wSize	Integer	Fixed	3	3/4	
wIntfSize	Integer	Fixed	2	2/3	
wTOF	Integer	Fixed	3	1/3	
wQoS	Integer	Derived	4	0 or 4	wQos = useQos ? 4 : 0;
wTNType	Integer	Fixed	8	8	
wAddr	Integer	Derived			From NC_ConcertoCSMIParams.json
wMPF1	Integer	Derived		8/12	From NC_ConcertoCSMIParams.json
wMPF2	Integer	Derived		6/12	From NC_ConcertoCSMIParams.json
wMPF3	Integer	Derived		5/12	From NC_ConcertoCSMIParams.json
wDId	Integer	Derived			From NC_ConcertoCSMIParams.json
wRBID	Integer	Derived			From NC_ConcertoCSMIParams.json ??
wRType	Integer	Fixed	1	1	
wNdpAux	Integer	Derived		0/16	Derivation is in mapping code = max {ArUser, AwUser}

TABLE 19-5: CONCERTOCPARAM PARAMETERS

Parameter Name	Type	Origin	Default	Min/ Max	Description/Derivation

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	5/10	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

19.3. ConcertoC RequestMessageFields

TABLE 19-6: 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;
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;
		}
		}

Table 19-7: ConcertoCRequestMessageFields Parameters

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;
		        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	

Table 19-8: ConcertoCRequestMessageFields Parameters

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;
		}
		}

Table 19-9: ConcertoCRequestMessageFields Parameters

Parameter Name	Type	Description/Derivation

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;
		    }
		}

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;
		    }
		}

TABLE 19-10: CONCERTOCREQUESTMESSAGEFIELDS PARAMETERS

Parameter Name	Type	Description/Derivation

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;
		}
		}

19.4. ConcertoCResponseMessageFields

TABLE 19-11: 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
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
		    }
		}

TABLE 19-12: CONCERTOCRESPONSEMESSAGEFIELDS PARAMETERS

Parameter Name	Type	Description/Derivation

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;
		}
		}

TABLE 19-13: CONCERTOCRESPONSEMESSAGEFIELDS PARAMETERS

Parameter Name	Type	Description/Derivation

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;
		}
		}

TABLE 19-14: CONCERTOCRESPONSEMESSAGEFIELDS PARAMETERS

Parameter Name	Type	Description/Derivation

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;
		}
		}

20. *Legato Derived/Fixed Parameters*

20.1. 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.

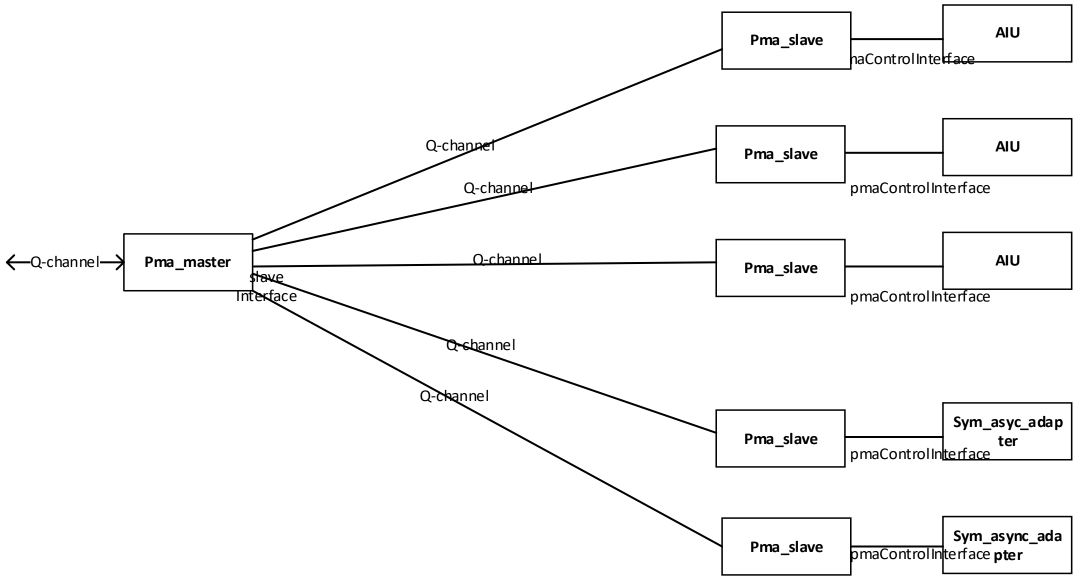


FIGURE 20-1: PMA IN CLOCK DOMAIN

20.2. Sym_async_adapter

TABLE 20-1: SYM_ASYNC_ADAPTER

Parameter Name	Default	Ranges	NCore 3.6	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.

20.3. Sym_buf_switch

All parameters for this element are not GUI visible

TABLE 20-2: 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

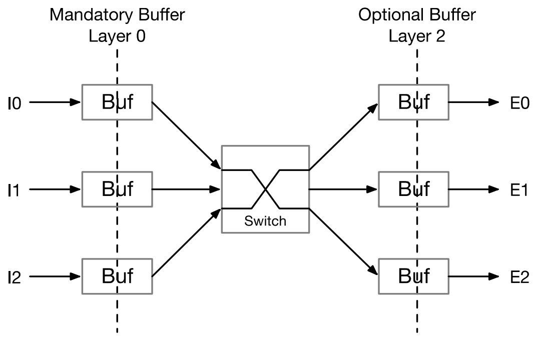


Figure 20-2: Sym_Buf_Switch in CDTI, with only one VC

20.3.1. 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.

20.4. Sym_ibuf_switch

NOTE: sym_ibuf_switch is required to support mesh topology.

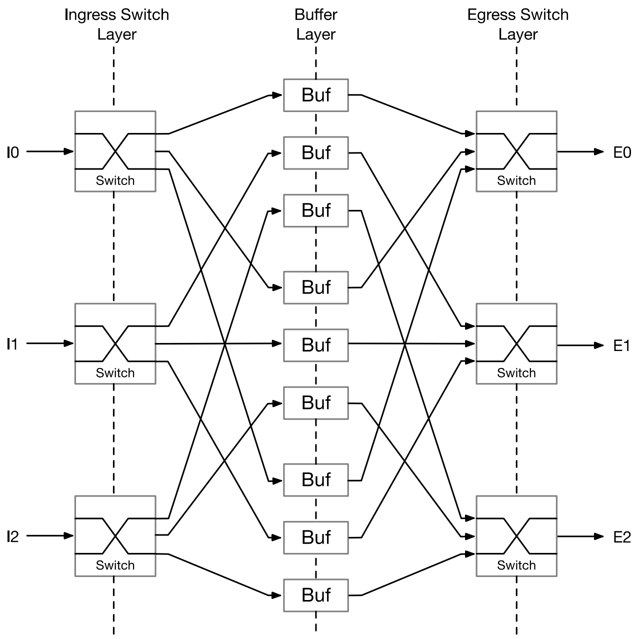


Figure 20-3: Sym_Buf_Switch in CDTI

TABLE 20-3: SYM_IBUF_SWITCH

Parameter Name	Default	Ranges	NCore 3.2	GUI-Visibility
arbType.egress	arb_rr1	arb_rr1, arb_pri_rr1, arb_fifo	Fixed	No
Circular	False	True/False	Derived	No
numPri	1		Derived	No

20.5. 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 20-4: NINPUTWIDTH PARAMETERS FOR WIDTHADAPTER

Name: nInputWidth					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	64	256	64	256	
Constraints	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.				

TABLE 20-5: NOUTPUTWIDTH PARAMETERS FOR WIDTHADAPTER

Name: nOutputWidth					Visibility: Engg
	Architecture		**Release**		**Default**
	Min	*Max*	*Min*	*Max*	
Value	64	256	64	256	
Constraints	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.				

Table 20-6: boolPipeline parameters for WidthAdapter

Name: boolPipeline					Visibility: User
	Architecture		*Release*		*Default*
Value	*True*	*False*	*True*	*False*	*False*
Constraints	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.				

Table 20-7: nDepth parameters for WidthAdapter

Name: nDepth					Visibility: Engg
	Architecture		*Release*		*Default*
	Min	*Max*	*Min*	*Max*	
Value	*0*	*4 x nTxnSize / nOutputWidth*	*0*	*4 x nTxnSize / nOutputWidth*	
Constraints	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					

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 20-8: NWIDTH PARAMETERS FOR RATEADAPTER

Name: nWidth					Visibility: Engg
	Architecture		*Release*		*Default*
	Min	*Max*	*Min*	*Max*	
Value	*64*	*256*	*64*	*256*	
Constraints	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				

TABLE 20-9: NDEPTH PARAMETERS FOR WIDTHADAPTER

Name: nDepth					Visibility: User
	Architecture		*Release*		*Default*
	Min	*Max*	*Min*	*Max*	
Value	*0*	*4 x nTxnSize / nOutputWidth*	*0*	*4 x nTxnSize / nOutputWidth*	
Constraints	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					

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 20-10: 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				

20.6. Sym_pipe_adpater

NOTE: chapter 16.6

TABLE 20-11: 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	

20.7. Interrupt

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

20.8. Parameter for CSR network

In the CSR network only atui_axi, 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.

20.9. 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.)

20.10. CSR fixed parameters

This chapter describes fixed parameters for CSR network

20.10.1. Atut_apb parameters

TABLE 20-12: 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	
maxOutTotal	1		Fixed as 1	
maxOutWr	1		Fixed as 1	

Parameter	Default	Ranges	CSR network	Description
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	

20.10.2. Atui_axi parameters

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

TABLE 20-13: 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

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

20.10.3. APB socket parameters

TABLE 20-14: 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	

20.10.4. AXI socket parameters

TABLE 20-15: AXI SOCKET PARAMETERS

Parameter Name	Default	Range	CSRnetwork	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	Fixed at 3	3	
wLock	1	AXI3: 2 [1:0] AXI4: 1	1	Exclusive accesses are always supported by default
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	

20.10.5. 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 20-16: 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

21. Other User Settable Parameters

21.1. Parameter related with Placeholder Generic Signal

Table 21-1: Parameter for Generic Port: wireName

Name: wire**Name**					Visibility: User
	Architecture		**Release**		**Default**
Value					
Constraints					
Customer Description	portName for Generic port				
Engineering Description					

Table 21-2: Parameter for Generic Port: wireWidth

Name: wireWidth					Visibility: User
	Architecture		**Release**		**Default**
Value					
Constraints					
Customer Description	Port width for generic port				
Engineering Description					

Table 21-3: Parameter for Generic Port: wireRtlPrefix

Name: **wireRtlPrefix**					Visibility: User
	Architecture		**Release**		**Default**
Value					
Constraints					
Customer Description	RTL Prefix. For a given block, all the ports must have the same witeRtlPrefix.				
Engineering Description					

Table 21-4: Parameter for Generic Port: direction

Name: direction					Visibility: User
	Architecture		**Release**		**Default**
	Valid Values		*Valid Values*		
Value	[In, Out]				In
Constraints					
Customer Description	Parameter for port direction configuration				
Engineering Description					

21.2. Parameter related with SRAM assignment

21.2.1. SW_memory

Table 21-5: Memory parameter for IOAIU

Memory Name	Constraints	Number of Memories
OttMem	MemoryProtectionType (Table 4-18) cannot be “NONE” memoryType must be SRAM	Same as #.of nOttDataBanks (Chapter 14.1)

Table 21-6: Memory parameter for Snoop Filter

Memory Name	Constraints	Number of Memories
TagMem	MemoryProtectionType (Table 4-18) cannot be “NONE” memoryType must be SRAM	Same as #.of nWays (Chapter 10.3) -- bitEn == 0 in NCore 3.2.

Table 21-7: 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 (Table 4-18) cannot be “NONE”, memoryType must be SRAM	1

Table 21-8: Memory parameter for DMI

Memory Name	Constraints	Number of Memories
writeDataMem	MemoryProtectionType (Table 4-18) cannot be “NONE” memoryType must be SRAM	Same as # of nCohWrDataBanks (Chapter 11.1)
rdDataMem	MemoryProtectionType (Table 4-18) 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 (Table 4-18) 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 (Table 4-18) cannot be “NONE”, memoryType must be SRAM	1

Table 21-9: Memory parameter for CCP

Memory Name	Constraints	Number of Memories
TagMem	MemoryProtectionType (Table 4-18) cannot be “NONE” memoryType must be SRAM	Same as #.of nTagBanks
DataMem	MemoryProtectionType (Table 4-18Table 4-19) cannot be “NONE” memoryType must be SRAM	Same as #.of nDataBanks

TABLE 21-10: MEMORY PARAMETER FOR DVE

Memory Name	Constraints	Number of Memories
TraceMem	MemoryProtectionType (Table 4-18) cannot be "NONE" MemoryType must be SRAM	2

TABLE 21-11: 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 (Table 4-18) cannot be "NONE", memoryType must be SRAM	1

TABLE 21-12: ENHALFSPEEDDATASRAM PARAMETER

Name: enHalfSpeedDataSRAM			Type: Int	Visibility: user Settable	
	Architecture		**Release**		**Default**
	Min	Max	Min	Max	
Value	0	1	0	1	0
Constraint	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				

TABLE 21-13: ENSRAMPIPE PARAMETER

Name: enSRAMPipe			Type: Int	Visibility: user Settable	
	Architecture		**Release**		**Default**
	Min	Max	Min	Max	
Value	0	1	0	1	0
Constraint	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.				
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.				

21.2.2. 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 21-14: PARAMETER FOR SRAM GENERIC PORT: WIRENAME

Name: wireName			Visibility: User
	Architecture	**Release**	**Default**
Value			
Constraints			
Customer Description	portName for Generic port		
Engineering Description			

TABLE 21-15: PARAMETER FOR SRAM GENERIC PORT: WIREWIDTH

Name: wireWidth			Visibility: User
	Architecture	**Release**	**Default**
Value		Max: 1024	
Constraints			
Customer Description	Port width for generic port Maximum width per signal in generic interface is 1024		
Engineering Description			

TABLE 21-16: PARAMETER FOR SRAM GENERIC PORT: DIRECTION

Name: direction			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	[In, Out]		In
Constraints			
Customer Description			
Engineering Description			

22. *User Settable parameter for Synthesis*

TABLE 22-1: SYNTHESIS PARAMETER: CHECKONLY

Name: checkOnly		**Type: Boolean**	Visibility: User
	Architecture	**Release**	**Default**
Value			
Constraints			
Customer Description	Whether to stop the RTL flow after compilation and linking, or to proceed to synthesis		
Engineering Description			

TABLE 22-2: SYNTHESIS PARAMETER: TOPOMODE

Name: topoMode		Type: Boolean	Visibility: User
	Architecture	**Release**	**Default**
Value			
Constraints			
Customer Description	Whether to launch the synthesis tools in topographical mode, or WLM. The latter is faster but less accurate		
Engineering Description			

TABLE 22-3: SYNTHESIS PARAMETER: TECHNOLOGY

Name: technode			Visibility: User
	Architecture	**Release**	**Default**
	Valid Values	*Valid Values*	
Value	ERROR,TSMC16,TSMC7,CUSTOM		CUSTOM
Constraints			
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			

TABLE 22-4: SYNTHESIS PARAMETER: CLOCKUNCERTAINTY

Name: c**lockUncertainty**					Visibility: User
	Architecture		**Release**		**Default**
	Min	*Max*			
Value	1	99			15
Constraints					
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					

TABLE 22-5: SYNTHESIS PARAMETER: RTLWRAPPERDIR

Name: rtlWrapperDir			Visibility: User
	Architecture	**Release**	**Default**
Value			
Constraints			
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			

Table 22-6: Synthesis Parameter: hardMacroDbs

Name: hardMacroDbs			Visibility: User
	Architecture	**Release**	**Default**
Value			
Constraints			
Customer Description	Specifies the location and names of the hard macros in the design, such as compiled memories		
Engineering Description			

Table 22-7: Synthesis Parameter: bottomUpSynthesis

Name: bottomUpSynthesis			Visibility: User
	Architecture	**Release**	**Default**
Value			True
Constraints			
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			

Table 22-8: Synthesis Parameter: maxTransition

Name: **maxTransition**			Visibility: User
	Architecture	**Release**	**Default**
Value			150
Constraints			
Customer Description	Default transition delay on functional input ports (THIS SHOULD BE RENAMED)		
Engineering Description			

Table 22-9: Synthesis Parameter: Output Load

Name: outputLoad			Visibility: User
	Architecture	**Release**	**Default**

Value			100000
Constraints			
Customer Description	The default capacitive load on functional output ports		
Engineering Description			

Table 22-10: Synthesis Parameter: ulvtPercentage

Name: ulvtPercentage			Visibility: User
	Architecture	**Release**	**Default**
Value			
Constraints			
Customer Description	Ulvt Percentage: Ulvt percentage limit set in synthesis scripts.		
Engineering Description			

Table 22-11: Synthesis Parameter: compileCommand

Name: compileCommand			Visibility: User
	Architecture	**Release**	**Default**
Value			
Constraints			
Customer Description	Compile command: Allows the user to add in options to default synthesis command		
Engineering Description			