Ncore Connectivity Architecture Specification

Rev: 0.87, February 16, 2023

ARTERIS® NCORE CONNECTIVITY ARCHITECTURE SPECIFICATION

Release Information

Version	*Editor*	*Change*	*Date*
0.5	MK	Initial Document	02/02/2022
O.6	MK	Updates after review	02/04/2022
0.7	MK	Updated interleaving optimization to include all bits after feedback from JP	03/01/2022
0.75	MK	Updated Table-13 with missing optimization row for 4way to 2 way interleaving Updated Table 9, Table 10 and Table 12 with missing connections	03/10/2022
0.76	MK	Added hexDceDmiVec parameter Fixed typos in message mapping table Clarification on what is included in CmdReq and CmdRsp	03/15/2022
0.80	MK	Added description of credit optimizations in DCE, involves both SW and HW changes	04/26/2022
0.82	MK	Added a SW check for DCE and DMI connectivity	05/05/2022
0.83	MK	Added limits to the capability of deleting routes between Ncore units Updated credit optimization section with new parameters to address implementation concerns Added SW rules for snoop filter Added description of possible SF optimization (not to be implemented) Increased RB credit limits from 16 to 32 AIU credit counter optimization	05/23/2022
0.84	MK	Added parameter for DCE connected DMIs	05/24/2022
0.85	MK	Updated mapping table with command response and update request/response details as per SW request	05/29/2022
0.86	MK	Added clarification for ACE-Lite E in the mapping table	06/01/2022
0.87	CCW	Add sysreq/sysrsp	02/16/2023
Legend:	MK MF CCW Xx	Mohammed Michael Frank Cheng Chung Wang Whoever else edited this document	

Confidentiality Status

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

Product Status

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

Web Address

http://www.arteris.com

Table of Contents

Table of Figures

Table of Tables

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.

Additional reading

This book contains information that is specific to this product. See the following documents for other relevant information.

History of the World II, Mel Brooks.

1 Introduction

This specification goes over Ncore transport interconnect connectivity of different Ncore units. It specifies how the connectivity must be optimized to take advantage of Ncore unit and port interleaving commonality. Furthermore, it also specifies when certain connectivity can be removed by choice and how Ncore units should report run time errors in these scenarios. The objective here is to optimize the network and remove any possible unused connectivity by the configuration.

1.1 Parameters

New parameters are introduced to specify port interleaving and to report connectivity information to AIUs. Parameters specified in Table 4, Table 5, Table 6, Table 7, Table 8, Table 9, Table 10 and Table 11 must be implemented as ports in RTL as tie offs.

Name: nAiuPorts **Type:** Int **Visibility:** User Settable

	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	1	16	1	4	1
Constraint	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 1, here it shows as a mutliported NCAIU with two AXI ports specified by nNativeInterfacePorts and then 2 NCAIUs specified by nAiuPorts.				

TABLE 1 NAIUPORTS PARAMETER

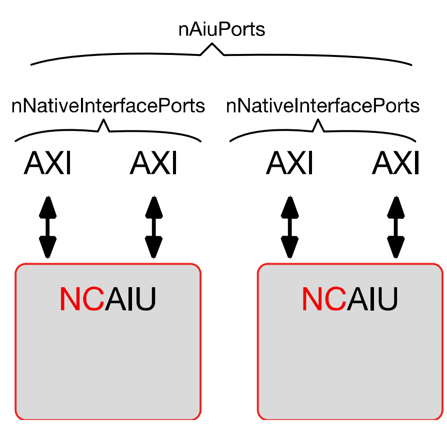


FIGURE 1 NAIUPORTS PARAMETER

Name: aPrimaryAiuPortBits		**Type:** array	**Visibility:** User Settable
	Architecture	**Release**	**Comment**
Parameters			
			array of integers
Constraint	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 2 APRIMARYAIUPORTBITS PARAMETER

Name: aSecondaryAiuPortBits		**Type:** array	**Visibility:** Engg
	Architecture	**Release**	**Comment**
Parameters	*Name*	*Name*	
			array of strings
Constraint	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	Note used in this release		

TABLE 3 ASECONDARYAIUPORTBITS PARAMETER

Name: hexAiuDceVec			**Type:** hex		**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 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 4 HEXAIUDCEVEC PARAMETER

Name: hexAiuDmiVec **Type:** hex **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 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 5 HEXAIUDMIVEC PARAMETER

Name: hexAiuDiiVec **Type:** hex **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 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 6 HEXAIUDIIVEC PARAMETER

Name: hexAiuConnectedDceFunitId **Type:** hex **Visibility:** Engg

	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	0	FFFFFFFF	0	FFFF	1
Constraint	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 FuntiIDs that are connected to the AIU				

TABLE 7 HEXAIUCONNECTEDDCEFUNITID PARAMETER

Name: hexDceConnectedDmiFunitId **Type:** hex **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 8 HEXDCECONNECTEDDMIFUNITID PARAMETER

Name: hexDceConnectedCaFunitId **Type:** hex **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 9 HEXDCECONNECTEDCAFUNITID PARAMETER

Name: hexDceDmiVec **Type:** hex **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 HEXDCEDMIVEC PARAMETER

Name: hexDceDmiRbOffset **Type:** hex **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 11 HEXDCEDMIRBOFFSET PARAMETER

Name: nAiuConnectedDces **Type:** Int **Visibility:** Engg

	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	1	64	1	32	1

Constraint	Number of DCEs connected to this each AIU.				
Customer Description					
Engineering Description	Specifies the number of caching agents (AIUs) that are connected to DCE				

TABLE 12 NAIUCONNECTEDDCES PARAMETER

Name: nDceConnectedCas **Type:** Int **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 13 NDCECONNECTEDCAS PARAMETER

Name: nDceConnectedDmis **Type:** Int **Visibility:** Engg

	Architecture		Release		Default
	Min	*Max*	*Min*	*Max*	
Value	1	32	1	16	1
Constraint	Number of DMIs connected to this 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 14 NDCECONNECTEDDMIS PARAMETER

Name: nDceRbCredits **Type:** Int **Visibility:** Engg

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

TABLE 15 NDCECONNECTEDDMIS PARAMETER

1.2 Connectivity mapping

This section specifies the full connectivity mapping without any optimizations.

CN0 is for control network 0, CN1 is for control network 1, CN2 is for control network 2 and DN is for data network. The mapping for the control and data network in 3CN1DN configuration is shown in Table 16.

2CN1DN configuration is achieved by combining CN1 and CN2

1CN1DN configuration is achieved by combining all the three CNs

Unit	Control Network 0 (CN0)		Control Network 1 (CN1)		Control Network 2 (CN2)		Data Network (DN)	
	Tx Msg	Rx Msg	Tx Msg	Rx Msg	Tx Msg	Rx Msg	Tx Msg	Rx Msg
CAIU - CHI								
	CmdReq	StrReq	StrRsp	CmdRsp			DtwReq/ DtwDbgReq	DtrReq
	SysReq	SnpReq	SnpRsp	DtrRsp			DtrReq	
		SysReq	DtrRsp	CmpRsp				
			SysRsp	DtwRsp/ DtwDbgRsp				
				SysRsp				
CAIU - ACE								
	CmdReq	StrReq	StrRsp	CmdRsp			DtwReq/ DtwDbgReq	DtrReq
	SysReq	SnpReq	SnpRsp	DtrRsp			DtrReq	
	UpdReq	SysReq	DtrRsp	CmpRsp				
			SysRsp	DtwRsp/ DtwDbgRsp				
				SysRsp				
				UpdRsp				
NCAIU (ACE-Lite/ACE-Lite E with DVM)								
	CmdReq	StrReq	StrRsp	CmdRsp			DtwReq/ DtwDbgReq	DtrReq
	SysReq	SnpReq	SnpRsp	DtrRsp (only ACE-Lite E)			DtrReq (only ACE-Lite E)	
			DtrRsp	DtwRsp/ DtwDbgRsp				
				SysRsp				
NCAIU (AXI with proxy cache)								
	CmdReq	StrReq	StrRsp	CmdRsp			DtwReq/ DtwDbgReq	DtrReq
	SysReq	SnpReq	SnpRsp	DtrRsp			DtrReq	
	UpdReq	SysReq	DtrRsp	DtwRsp/ DtwDbgRsp				
			SysRsp	SysRsp				
				UpdRsp				
NCAIU (AXI without proxy cache and ACE-Lite/ACE-Lite E without DVM)								
	CmdReq	StrReq	StrRsp	CmdRsp			DtwReq/ DtwDbgReq	DtrReq
			DtrRsp	DtrRsp (only ACE-Lite E)			DtrReq (only ACE-Lite E)	
				DtwRsp/ DtwDbgRsp				
DCE								
	StrReq	CmdReq	CmdRsp	SnpRsp	MrdReq	MrdRsp		
	SnpReq	SysReq	SysRsp	StrRsp	RbrReq	RbrRsp		
	SysReq	UpdReq	UpdRsp	SysRsp	RbuRsp	RbuReq		

Unit	Control Network 0 (CN0)		Control Network 1 (CN1)		Control Network 2 (CN2)		Data Network (DN)	
	Tx Msg	Rx Msg	Tx Msg	Rx Msg	Tx Msg	Rx Msg	Tx Msg	Rx Msg
DII								
	StrReq	CmdReq	NCCmdRsp	StrRsp			DtrReq	DtwReq
			DtwRsp	DtwDbgRsp			DtwDbgReq	
				DtrRsp				
DMI								
	StrReq	CmdReq	NCCmdRsp	StrRsp	MrdRsp	MrdReq	DtrReq	DtwReq
			DtwRsp	DtwDbgRsp	RbrRsp	RbrReq	DtwDbgReq	
				DtrRsp	RbuReq	RbuRsp		
DVE								
	StrReq	CmdReq	NCCmdRsp	StrRsp				DtwReq/ DtwDbgReq
	SnpReq	SysReq	CmpRsp	SnpRsp				
	SysReq		DtwRsp/ DtwDbgRsp	SysRsp				
			SysRsp					

TABLE 16 SYSTEM CONTROL AND DATA NETWORK MAPPING

Connectivity between different Ncore units is shown in Table 17. The top row and the left most column specify the Ncore unit names, the intersection point specifies the networks that connect the two units. Intersection points that are empty signifies that there are no connections between the corresponding units.

	CAIU	NCAIU	DCE	DII	DMI	DVE
CAIU	DN CN1	DN CN1	CN0 CN1	CN0 CN1 DN	CN0 CN1 DN	CN0 CN1 DN
NCAIU	DN CN1	DN CN1	CN0 CN1	CN0 CN1 DN	CN0 CN1 DN	CN0 CN1 DN
DCE	CN0 CN1	CN0 CN1			CN2	
DII	CN0 CN1 DN	CN0 CN1 DN				
DMI	CN0 CN1 DN	CN0 CN1 DN	CN2			
DVE	CN0 CN1 DN	CN0 CN1 DN				

TABLE 17 CONNECTIVITY MAPPING

Detailed TX and RX connectivity between Ncore units for different networks are shown in Table 18, Table 19, Table 20 and Table 21. The top row and the left most column specify the Ncore unit names, the intersection point specifies the connectivity. The direction of connectivity is from column name perspective, i.e. the port names TX/RX specifies the connectivity on the column unit.

	CAIU CHI	CAIU ACE	NCAIU AXI W Cache	NCAIU AXI No Cache	NCAIU ACE-Lite	NCAIU ACE-Lite-E	DCE	DII	DMI	DVE
CAIU CHI							TX/RX	TX/RX	TX/RX	TX/RX
CAIU ACE							TX/RX	TX/RX	TX/RX	TX/RX
NCAIU AXI W Cache							TX/RX	TX/RX	TX/RX	TX/RX
NCAIU AXI No Cache							TX/RX	TX/RX	TX/RX	
NCAIU ACE-Lite							TX/RX	TX/RX	TX/RX	TX/RX(if DVM both)
NCAIU ACE-Lite-E							TX/RX	TX/RX	TX/RX	TX/RX(if DVM both)
DCE	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX				
DII	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX				
DMI	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX				
DVE	TX/RX	TX/RX	TX/RX		TX/RX (if DVM both)	TX/RX (if DVM both)				

TABLE 18 CN0 TX RX CONNECTIVITY MAP

	CAIU CHI	CAIU ACE	NCAIU AXI W Cache	NCAIU AXI No Cache	NCAIU ACE-Lite	NCAIU ACE-Lite-E	DCE	DII	DMI	DVE
CAIU CHI	TX/RX	TX/RX	TX/RX	TX	TX	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX
CAIU ACE	TX/RX	TX/RX	TX/RX	TX	TX	TX	TX/RX	TX/RX	TX/RX	TX/RX
NCAIU AXI W Cache	TX/RX	TX/RX	TX/RX	TX	TX	TX	TX/RX	TX/RX	TX/RX	TX/RX
NCAIU AXI No Cache	RX	RX	RX				TX/RX	TX/RX	TX/RX	TX
NCAIU ACE-Lite	RX	RX	RX				TX/RX	TX/RX	TX/RX	TX/RX(if DVM)
NCAIU ACE-Lite-E	TX/RX	RX	RX				TX/RX	TX/RX	TX/RX	TX/RX(if DVM)
DCE	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX				
DII	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX				TX
DMI	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX				TX
DVE	TX/RX	TX/RX	TX/RX	RX	TX(if DVM)/RX	TX(if DVM)/RX		RX	RX	

TABLE 19 CN1 TX RX CONNECTIVITY MAP

	CAIU CHI	CAIU ACE	NCAIU AXI W Cache	NCAIU AXI No Cache	NCAIU ACE-Lite	NCAIU ACE-Lite-E	DCE	DII	DMI	DVE
CAIU CHI										
CAIU ACE										
NCAIU AXI W Cache										
NCAIU AXI No Cache										
NCAIU ACE-Lite										
NCAIU ACE-Lite-E										
DCE									TX/RX	
DII										
DMI							TX/RX			
DVE										

TABLE 20 CN2 TX RX CONNECTIVITY MAP

	CAIU CHI	CAIU ACE	NCAIU AXI W Cache	NCAIU AXI No Cache	NCAIU ACE-Lite	NCAIU ACE-Lite-E	DCE	DII	DMI	DVE
CAIU CHI	TX/RX	TX/RX	TX/RX	RX	RX	TX/RX		TX/RX	TX/RX	RX
CAIU ACE	TX/RX	TX/RX	TX/RX	RX	RX	RX		TX/RX	TX/RX	RX
NCAIU AXI W Cache	TX/RX	TX/RX	TX/RX	RX	RX	RX		TX/RX	TX/RX	RX
NCAIU AXI No Cache	TX	TX	TX					TX/RX	TX/RX	RX
NCAIU ACE-Lite	TX	TX	TX					TX/RX	TX/RX	RX
NCAIU ACE-Lite-E	TX/RX	TX	TX					TX/RX	TX/RX	RX
DCE										
DII	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX				RX
DMI	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX	TX/RX				RX
DVE	TX	TX	TX	TX	TX	TX		TX	TX	

TABLE 21 DN TX RX CONNECTIVITY MAP

1.3 Connectivity Optimization

This section goes over connectivity optimizations that must be applied to remove connections from the full connectivity described in Connectivity mapping section.

1.3.1 Removing connectivity

Depending on the system requirements a customer may choose to design a system where only a subset of AIUs may talk to only a subset of DIIs, this includes CSR configuration DII. Following is required

SW requirements:

- Provide GUI/TCL way to delete connections (all routes for all messages) between any AIUs and DIIs with following restriction
  - If an AIU is specified as CSR access capable AIU where parameter fnCsrAccess is set, then the customer must not be able to delete connection to CSR configuration DII
- Make sure the parameter hexAiuDiiVec is set correctly once the connections are deleted

HW requirments

- If a transaction gets decoded to a DII to which the AIU is not connected based on port tie offs associated with parameters hexAiuDiiVec; then report an error as address decode error with additional information as specified for error type code 0x7

1.3.2 AIU – DCE Connectivity Optimization

In cases where interleaving address bit commonality is present between an AIU group (AIU ports marked as interleaved) and DCE interleaving then the connection between them must be optimized.

- If interleaving granularity is same and all address bits match, then optimization must be implemented i.e. connectivity becomes one to one
- If interleaving granularity is same and necessarily all address bits do not match but at-least one or more bit at the same index match, then optimization must be implemented. The level optimization depends on number of address bits that match.
- If interleaving granularity is same and all address bits do not match, then no optimization is required

Detailed examples for different combinations of 2-way and 4-way interleaving are shown in Table 22. In the example AIUs and DCEs are numbered as follows for 4-way interleaving

- 00 → AIU0, DCE0
- 01 → AIU1, DCE1
- 10 → AIU2, DCE2
- 11 → AIU3, DCE3

Scenario	AIU interleaving address bits	DCE interleaving address bits	Optimization	Comment
2-way interleaved AIU and 2-Way interleaved DCE	X	X	AIU0 → DCE0 AIU1 → DCE1	One to one completely optimized connectivity
	X	Y	AIU0 → DCE0, DCE1 AIU1 → DCE0, DCE1	No optimization complete cross bar
4-way interleaved AIU and 4-way interleaved DCE	X, Y	X, Y	AIU0 → DCE0 AIU1 → DCE1 AIU2 → DCE2 AIU3 → DCE3	One to one completely optimized connectivity
	X, Y	X, A	AIU0 → DCE0, DCE1 AIU1 → DCE0, DCE1 AIU2 → DCE2, DCE3 AIU3 → DCE2, DCE3	Partially 1 to 2 optimized as the MSB bit is same and the LSB bit is not same
	X, Y	A, Y	AIU0 → DCE0, DCE2 AIU1 → DCE1, DCE3 AIU2 → DCE0, DCE2 AIU3 → DCE1, DCE3	Partially 1 to 2 optimized as the LSB bit is same and the MSB bit is not same
	X, Y	A, B	AIU0 → DCE0, DCE1, DCE2, DCE3 AIU1 → DCE0, DCE1, DCE2, DCE3 AIU2 → DCE0, DCE1, DCE2, DCE3 AIU3 → DCE0, DCE1, DCE2, DCE3	Complete cross bar, both bits are different.
	X	X, A	AIU0 → DCE0, DCE1 AIU1 → DCE2, DCE3	Partially 1 to 2 optimized as the MSB bit is same

Scenario	AIU interleaving address bits	DCE interleaving address bits	Optimization	Comment
2-way interleaved AIU and 4-Way interleaved DCE	X	A, X	AIU0 → DCE0, DCE2 AIU1 → DCE1, DCE3	Partially 1 to 2 optimized as the LSB bit is same
	X	A, B	AIU0 → DCE0, DCE1, DCE2, DCE3 AIU1 → DCE0, DCE1, DCE2, DCE3	No optimization complete cross bar
4-way interleaved AIU and 2-Way interleaved DCE	X, Y	X	AIU0 → DCE0 AIU1 → DCE0 AIU2 → DCE1 AIU3 → DCE1	2 to one optimization as the MSB bit matches
	X, Y	Y	AIU0 → DCE0 AIU1 → DCE1 AIU2 → DCE0 AIU3 → DCE1	2 to one optimization as the LSB bit matches
	X, Y	A	AIU0 → DCE0, DCE1 AIU1 → DCE0, DCE1 AIU2 → DCE0, DCE1 AIU3 → DCE0, DCE1	No optimization complete cross bar 1) only LSB bit matches 2) none of the bits match

TABLE 22 AIU DCE INTERLEAVING EXAMPLE

SW requirements:

- Make sure the parameter hexAiuDceVec is set correctly once the connections are optimized
- Compute nDceConnectedCas and nAiuConnectedDces

HW requirements

- If a transaction gets decoded to a DCE to which the AIU is not connected based on port tie offs associated with parameter hexAiuDceVec; then report an error as address decode error with additional information as specified for error type code 0x7

1.3.3 AIU/DCE – DMI Connectivity Optimization

In cases where interleaving address bit commonality is present between an AIU group (AIU ports marked as interleaved) / DCEs and DMI interleaving, then the connection between them must be optimized.

- If interleaving granularity is same and all address bits match, then optimization must be implemented i.e. connectivity becomes one to one
- If interleaving granularity is same and necessarily all address bits do not match but at-least one or more bit at the same index match, then optimization must be implemented. The level optimization depends on number of address bits that match.
- If interleaving granularity is same and all address bits do not match, then no optimization is required

In the case of DMI commonality must be considered across the different interleaving options specified, if there is an intersection then optimization can be done if not then no optimization applies.

SW requirements:

- Make sure the parameters hexAiuDceVec, hexAiuDmiVec and hexDceDmiVec are set correctly once the connections are optimized
- Compute nDmiConnectedDces and nDceConnectedDmis

HW requirements

- If a transaction gets decoded to a DCE/DMI to which the AIU/DCE is not connected based on port tie offs associated with parameters hexAiuDceVec, hexAiuDmiVec and hexDceDmiVec; then report an error as address decode error with additional information as specified for error type code 0x7

1.3.4 AIU – AIU Connectivity Optimization

In cases where interleaving address bit commonality is present between interleaved AIUs, then the connection between them must be optimized based on following:

- AIUs within a single interleaved group must have all connectivity between them optimized.
- AIUs across interleaved groups; if the interleaved address bit/s are not same for any two or more AIUs then all connectivity between these AIUs must be optimized. This is irrespective of the granularity of interleaving

Detailed Example is shown in Figure 1.

In group 0, AIU0 and AIU1 are 2 way interleaved with bit 7. As they are within a single group, they do not have any connectivity between them.

In group 1, AIU2, AIU3, AIU4 and AIU5 are 4 way interleaved with bit 7,8. As they are within a single group, they do not have any connectivity between them.

As there is commonality between the two groups, connections are optimized as shown where AIU0 talks with AIU2 and AIU3 as they share the interleaving bit 7 with a value of 0, same applies between AIU1 and AIU4, AIU5.

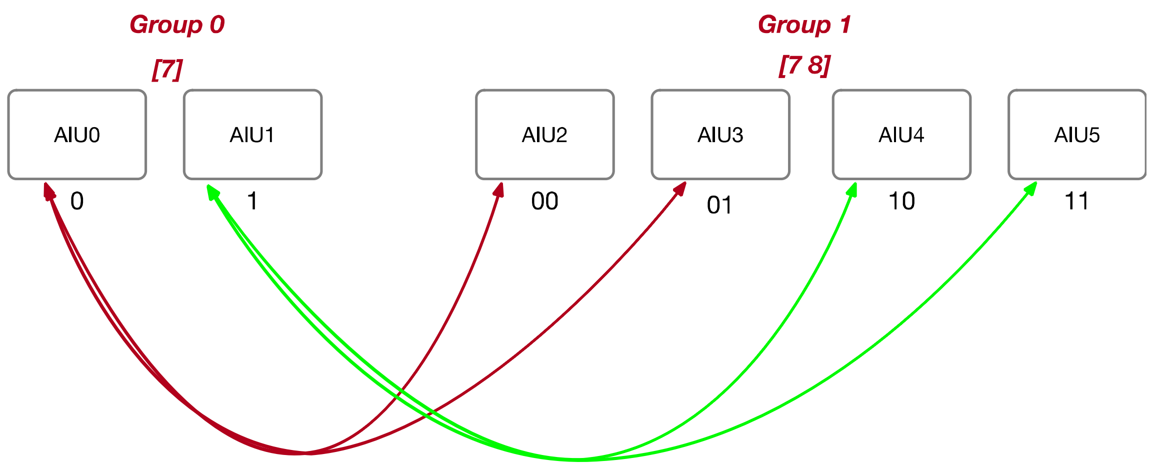


FIGURE 2 AIU TO AIU CONNECTIVITY

1.3.5 Snoop filter optimizations

The interleaving presents an opportunity to optimize snoop filter sharer vector. Details of possible optimization are discussed in Opens section. At this time these optimizations will not be implemented by Ncore 3, apart from following restrictions

1. Interleaved AIUs within a group must be assigned to the same Snoop filter. This restriction makes sense as it is expected that all agents connecting via interleaved ports within a group will have the same or shared cache structure.
2. Number of ways within a snoop filter must be limited to multiple of 4, with a max value of 32. This restriction is to reduce possible combinations.

1.4 Credit Optimization

This section goes over credit optimizations that must be implemented to compliment the connectivity optimizations. Credits in question here are build time defined credits specified for DCE i.e., nDceRbCredits and nAiuSnpCredits. Other credits do not get affected as they will be SW defined at run time. Credit parameter specification stays the same with following changes in software and hardware.

SW changes:

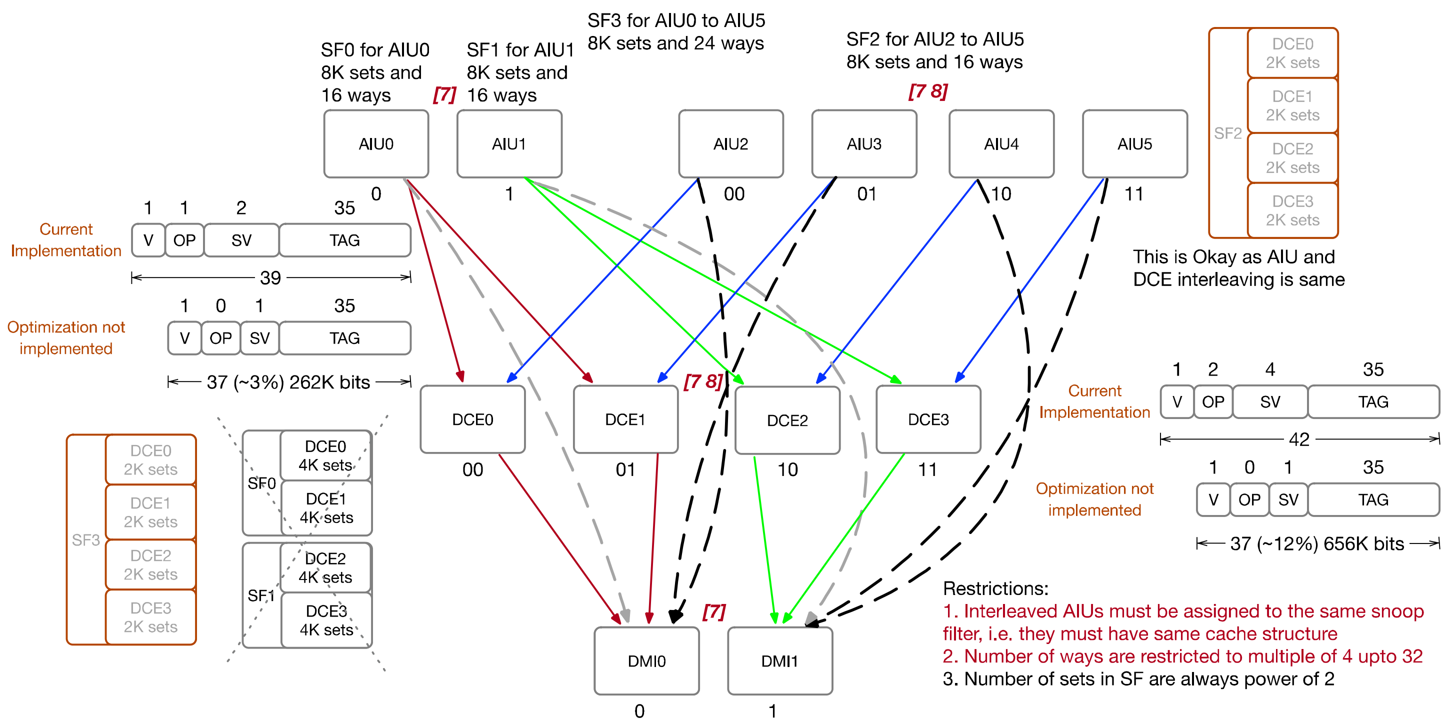
- nDceRbCredits are used to derive the coherent write buffer size in DMI. The size of this buffer must be limited to the sum of nDceRbCredits of only the DCEs that are connected to the DMI.
- nAiuSnpCredits are used to derive the depth of the STT table (nSttCtrlEntries) in AIUs. The size of this STT table must be limited to sum of nAiuSnpCredits of only the DCEs that are connected to the AIU

HW changes:

- Implement RbCredit counters up to the number specified by the parameter nDceConnectedDmis. The offset value for Rb credits must be used from the DCE port tie off parameter hexDceDmiRbOffset.
- Implement snoop credit counters up to the number specified by the parameter nDceConnectedCas. (This is optional, we can defer it to reduce RTL change, note that this change may help in timing)

Note: AIUs have command credits counters, these can be optimized based on actual targets (DCEs, DMIs, DIIs) connected to the AIU.

2 Opens


SF3 for AIU0 to AIU5
8K sets and 24 ways
SF0 for AIU0
8K sets and
16 ways
[7]
SF1 for AIU1
8K sets and
16 ways
SF2 for AIU2 to AIU5
8K sets and 16 ways
[7 8]
AIU0
AIU1
AIU2
AIU3
AIU4
AIU5
0
1
00
01
10
11
DCE0
2K sets
DCE1
2K sets
SF2
DCE2
2K sets
DCE3
2K sets
This is Okay as AIU and
DCE interleaving is same
Current
Implementation
1 1 2 35
V OP SV TAG
39
Optimization not
implemented
1 0 1 35
V OP SV TAG
37 (~3%) 262K bits
DCE0
DCE1
[7 8]
DCE2
DCE3
00
01
10
11
Current
Implementation
1 2 4 35
V OP SV TAG
42
Optimization not
implemented
1 0 1 35
V OP SV TAG
37 (~12%) 656K bits
SF3
DCE0
2K sets
DCE1
2K sets
DCE2
2K sets
DCE3
2K sets
SF0
DCE0
4K sets
DCE1
4K sets
SF1
DCE2
4K sets
DCE3
4K sets
[7]
DMI0
DMI1
0
1
Restrictions:
1. Interleaved AIUs must be assigned to the same snoop
filter, i.e. they must have same cache structure
2. Number of ways are restricted to multiple of 4 upto 32
3. Number of sets in SF are always power of 2

3 Glossary

Arteris

A NoC Company

NCore3

A coherent NoC provided by Arteris with AMBA interfaces and built-in caches.

4 Notes

Notes ……