Ncore Error Architecture Specification

Release: 3.4

Rev: 0.89,

February 24, 2023

Release Information

Version	*Editor*	*Change*	*Date*
0.5	MK	Initial Document with 3.4 release updates	02/04/2022
0.6	MK	Added clarification on Address decode error Added the need for DCE to drop the recall transaction	03/31/2022
0.7	MK	Added clarification on ACE snoop response error snoop filter update Added clarification on CHI Error response flits	06/17/2022
0.8	MK	Added reporting for NS miss match on Address map	09/07/2022
0.82	MK	Added in band error response for DCE no credits error	09/08/2022
0.84	MK	Updated typo in address error code for ACE CR Added DCE StrResp propagation on credit error Added DVM error propagation	11/18/2022
0.85	MF	Modified 1.2.3.2 Section: Uncorrectable ErrorType and Info Table Cleaned up some table formatting	12/16/2022
0.86	MF	Added information about SRAM address protection - CONC-11081	01/24/2023
0.87	MF/CCW	Added clarification for atomic transaction errors	01/27/2023
0.88	MF	Updated SysCo error in section 1.2.3.2 to indicate external error arriving from system response Add software programming error of atomic transactions - CONC-11372	02/09/2023
0.89	CCW	Add clarification of dataless transactions with data buffer error	02/24/2023
Legend:	MK	Mohammed	
	MF	Michael Frank	
	CCW	Cheng Chung Wang	
	Xx	Whoever else edited this document	

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.

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

NO TABLE OF FIGURES ENTRIES FOUND.

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.

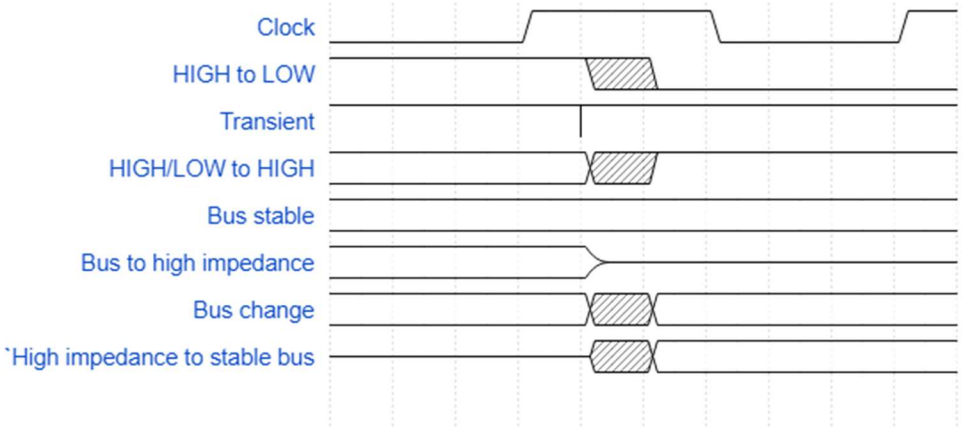
SMALL CAPITALS

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 Error handling in Ncore. This spec is derived from current implementation of Ncore 3.0 which was partially derived from Ncore 2.x. Effort has been made to reduce the number of changes from current implementation. In a later version of Ncore more changes and improvements will be introduced. The document first describes different registers and encodings and then goes into details of how these errors are handled

1.1 Parameters

No new parameters introduced.

1.2 Error registers

Refer to CSR CPRs in in the hw-ncr repository under cpr/csr for register addresses.

1.2.1 Correctable Error Registers

1.2.1.1 Correctable Error Control Register (xCECR)

Register Description: This register controls the detection and interrupt signaling of Correctable Errors in the unit.

Register Fields:

Bits	Name	Access	Reset	Description
0	ErrDetEn	RW	0b0	Correctable Error Detection Enable. When set to 1, this bit enables detection and logging of correctable errors.
1	ErrIntEn	RW	0b0	Correctable Error Interrupt Enable. When set to 1, this bit enables the assertion of Correctable Error Interrupt signal when a new Correctable Error is detected and ErrCount value equals to ErrThreshold.
3:2	Reserved	-	-	-
11:4	ErrThreshold	RW	0x00	Correctable Error Threshold. Number of corrected errors for which logging, and interrupt (if ErrIntEn is set to 1) is suppressed.
31:12	Reserved	-	-	-

1.2.1.2 Correctable Error Status Register (xCESR)

Register Description: This register logs information of Correctable Errors in the unit. If a new Correctable Error attempts to log this register the same cycle xCESAR is configured, this register will contain information from xCESAR.

Register Fields:

Bits	Name	Access	Reset	Description
0	ErrVld	W1C	0b0	Correctable Error Valid. This bit is set when a Correctable Error is detected and ErrCount equals to ErrThreshold. xCESR, xCELR0 and xCELR1 contain logged information about this Correctable Error. Write 1 to the field resets it to 0b0.
1	ErrCountOverflow	RO	0b0	Correctable Error Count Overflow. This bit is set to 1 when ErrVld is asserted, and a new Correctable Error is detected. Once the field is set, it will keep asserted until being reset. The field is reset to 0b0 when ErrVld is reset.
9:2	ErrCount	RO	0x00	Correctable Error Count. This field increments when a Correctable Error is detected. Once the field reaches ErrThreshold, the value will be frozen from that time until being reset. The field is reset to 0x00 when ErrVld is reset. If write 1 to ErrVld and a new Correctable Error is detected at the same cycle, the field will not increment for the error.
11:10	Reserved	-	-	-
15:12	ErrType	RO	0x00	Correctable Error Type. When ErrVld is set to 1, the field indicates the type of error that has been detected and logged.
31:16	ErrInfo	RO	0x0000	Correctable Error Information. When ErrVld is set to 1, the field contains additional unit-specific and error-type-specific information about the error. For example, information that could help identify the site of the error.

1.2.1.3 Correctable Error Location Register (xCELR0)

Register Description: This register logs location information of Correctable Errors in the unit. The information is unit-specific (see Error Type and Information Summary). The fields are made RW accessible to utilize this register as the alias register for error information injection purpose.

Register Fields:

Bits	Name	Access	Reset	Description
19:0	ErrEntry	RW	0x00000	Error Entry. When ErrVld is set to 1, the field contains location information of the error. If the error is from a memory array, it indicates the Entry or Set (if from a cache) number where the error occurs. Otherwise, it is the [19:0] bits of the error address.
25:20	ErrWay	RW	0x00	Error Way. When ErrVld is set to 1, the field contains location information of the error. If the error is from a cache, it indicates the Way number where the error occurs. Otherwise, it is the [25:20] bits of the error address.
31:26	ErrWord	RW	0x00	Error Word. When ErrVld is set to 1, the field contains location information of the error. If the error is from a cache, it indicates the beat number where the error occurs. Otherwise, it is the [31:26] bis of the error address.

1.2.1.4 Correctable Error Location Register (xCELR1)

Register Description: This register logs extra address information that cannot fit in xCELR0. The fields are made RW accessible to utilize this register as the alias register for error information injection purpose.

Register Fields:

Bits	Name	Access	Reset	Description
19:0	ErrAddr	RW	0x000	Error Address Higher Bits. Contains higher bit of error address if xELR0 cannot fit the whole error address.
31:20	Reserved	-	-	-

1.2.1.5 Correctable Error Status Alias Register (xCESAR)

Register Description: This register is used to inject error status into xCESR. If a new Correctable Error attempts to log the same cycle this register is configured, the information in this register will be logged in the xCESR.

Register Fields:

Bits	Name	Access	Reset	Description
0	ErrVld	RW	0b0	Alias bit for setting ErrVld in xCESR.
1	ErrCountOverflow	RW	0b0	Alias bit for setting ErrCountOverflow in xCESR.
9:2	ErrCount	RW	0x00	Alias field for setting ErrCount in xCESR.
11:10	Reserved	-	-	-
15:12	ErrType	RW	0x00	Alias field for setting ErrType in xCESR.
31:16	ErrInfo	RW	0x0000	Alias field for setting ErrInfo in xCESR.

1.2.1.6 Correctable Resiliency Threshold Register (xCRTR)

Register Description: This register contains the resiliency correctable error threshold as an indication to the functional safety controller (FSC). Exists only if the unit enables resiliency.

Register Fields:

Bits	Name	Access	Reset	Description
7:0	ResThreshold	RW	0x01	Resiliency Correctable Error Threshold. See FSC documents for more information.
31:8	Reserved	-	-	-

1.2.2 Uncorrectable Error Registers

1.2.2.1 Uncorrectable Error Interrupt Register (xUEDR)

Register Description: This register enables the detection of Uncorrectable Errors.

Register Fields:

Bits	Name	Access	Reset	Description
0	ProtErrDetEn	RW	0b0	Protocol Error Detection Enable. When set to 1, this bit enables the detection and logging of Protocol type Uncorrectable Error.
1	TransErrDetEn	RW	0b0	Transport Error Detection Enable. When set to 1, this bit enables the detection and logging of Transport type Uncorrectable Error.
2	MemErrDetEn	RW	0b0	Memory Error Detection Enable. When set to 1, this bit enables the detection and logging of Memory type Uncorrectable Error. This field exists only when there are protected memory arrays in the unit.
3	DecErrDetEn	RW	0b0	Decode Error Detection Enable. When set to 1, this bit enables the detection and logging of Access type Uncorrectable Error.
4	TimeOutErrDetEn	RW	0b0	Time Out Error Detection Enable. When set to 1, this bit enables the detection and logging of Time Out Error.
31:5	Reserved	-	-	-

1.2.2.2 Uncorrectable Error Interrupt Register (xUEIR)

Register Description: This register enables the interrupt signaling of Uncorrectable Errors.

Register Fields:

Bits	Name	Access	Reset	Description
0	ProtErrIntEn	RW	0b0	Protocol Error Interrupt Enable. When set to 1, this bit enables the assertion of Protocol type Uncorrectable Error Interrupt signal.
1	TransErrIntEn	RW	0b0	Transport Error Interrupt Enable. When set to 1, this bit enables the assertion of Transport type Uncorrectable Error Interrupt signal.
2	MemErrIntEn	RW	0b0	Memory Error Interrupt Enable. When set to 1, this bit enables the assertion of Memory type Uncorrectable Error Interrupt signal. This field exists only when there are protected memory arrays in the unit.
3	DecErrIntEn	RW	0b0	Decode Error Interrupt Enable. When set to 1, this bit enables the assertion of Access type Uncorrectable Error Interrupt signal.
4	TimeOutErrIntEn	RW	0b0	Time Out Error Interrupt Enable. When set to 1, this bit enables the assertion of Time Out Error Interrupt signal.
31:5	Reserved	-	-	-

1.2.2.3 Uncorrectable Error Status Register (xUESR)

Register Description: This register logs information about Uncorrectable errors in the unit. If a new Uncorrectable Error attempts to log this register the same cycle xUESAR is configured, this register will contain information from xUESAR.

Register Fields:

Bits	Name	Access	Reset	Description
0	ErrVld	W1C	0b0	Uncorrectable Error Valid. This bit is set when an Uncorrectable Error is detected. xUESR, xUELR0 and xUELR1 contain logged information about this Uncorrectable Error. Write 1 to the field resets it to 0b0. If write 1 to the field and a new Uncorrectable Error is detected at the same cycle, the field is cleared.
3:1	Reserved	-	--	-
7:4	ErrType	RO	0x00	Uncorrectable Error Type. When ErrVld is set to 1, the field indicates the type of error that has been detected and logged.
15:8	Reserved	-	-	-
31:16	ErrInfo	RO	0x0000	Uncorrectable Error Information. When ErrVld is set to 1, the field contains additional unit-specific and error-type-specific information about the error. For example, information that could help identify the site of the error.

1.2.2.4 Uncorrectable Error Location Register (xUELR0)

Register Description: This register logs location information of Uncorrectable Errors in the unit. The information is unit-specific (see Error Type and Information Summary). The fields are made RW accessible to utilize this register as the alias register for error information injection purpose.

Register Fields: see Correctable Error Location Register (xCELR0)

1.2.2.5 Uncorrectable Error Location Register (xUELR1)

Register Description: This register logs extra address information that cannot fit in xUELR0. The fields are made RW accessible to utilize this register as the alias register for error information injection purpose.

Register Fields: see

Correctable Error Location Register (xCELR1)

1.2.2.6 Uncorrectable Error Status Alias Register (xUESAR)

Register Description: This register is used to inject error status into xUESR. If a new Uncorrectable Error attempts to log the same cycle this register is configured, the information in this register will be logged in the xUESR.

Register Fields:

Bits	Name	Access	Reset	Description
0	ErrVld	RW	0b0	Alias bit for setting ErrVld in xUESR.
3:1	Reserved	-	-	-
7:4	ErrType	RW	0x00	Alias field for setting ErrType in xUESR.
15:8	Reserved	-	-	-
31:16	ErrInfo	RW	0x0000	Alias field for setting ErrInfo in xUESR.

1.2.3 Error Type and Information Summary

1.2.3.1 Correctable ErrorType and Info Table

ErrType Code	Error Type and ErrInfo Code	Error Location			
		Addr	Word	Way	Entry
0x0	Data Correctable Error • [2:0] – Storage Type ○ 3'b000 – OTT-Data ○ 3'b001 – Read Buffer ○ 3'b010 – Write Buffer ○ 3'b011 – Snoop Filter ○ 3'b100 – Trace Buffer • [7:3] – Reserved • [15:8] – Snoop Filter ID (Reserved for non-snoop filter)	MBZ*/ MBZ	MBZ/N/A	MBZ/Way	Entry/ Set
0x1	Cache Correctable Error • [0] – Array Type ○ 1'b0 - Tag Array ○ 1'b1 - Data Array	MBZ	Word	Way	Set
0x2 – 0xF	Reserved	N/A	N/A	N/A	N/A

*MBZ: Must Be Zero

1.2.3.2 Uncorrectable ErrorType and Info Table

ErrType Code	Error Type and ErrInfo Code	Error Location			
		Addr	Word	Way	Entry
0x0	Data Uncorrectable Error (applies to data and memory address errors) • [2:0] – Storage Type ○ 3'b000 – OTT-Data ○ 3'b001 – Read Buffer ○ 3'b010 – Write Buffer ○ 3'b011 – Snoop Filter ○ 3'b100 – Trace Buffer • [7:3] – Reserved • [15:8] – Snoop Filter ID (Reserved for non-snoop filter)	MBZ*/ MBZ	MBZ/N/ A	MBZ/W ay	Entry/ Set
0x1	Cache Uncorrectable Error (applies to tag/data SRAM and memory address errors) • [0] – Array Type ○ 1'b0 - Tag Array ○ 1'b1 - Data Array	MBZ	Word	Way	Set
0x2	Native Interface Write Response Error • [1:0] - Response • [2] - Security Attribute • [3]- Eviction	Transaction Address			
0x3	Native Interface Read Response Error • [1:0] - Response • [2] - Security Attribute • [3]- Fill	Transaction Address			
0x4	Native Interface Snoop Response Error • [1:0] – Response (Reserved for Snoop filters) • [2] - Security Attribute	Transaction Address			
0x5-0x6	Reserved	N/A	N/A	N/A	N/A
0x7	Decode Error • [3:0] – Type ○ 4'b0000: No address hit ○ 4'b0001: Multiple address hit ○ 4'b0010: Illegal CSR access format ○ 4'b0011: Illegal DII access type ○ 4'b0100: unconnected DMI unit access ○ 4'b0101: unconnected DII unit access ○ 4'b0110: unconnected DCE unit access ○ 4'b0111: No credits configured for access ○ 4'b1000: Illegal security access ○ 4'b10001 to 4'b1111: Reserved	Transaction Address			

ErrType Code	Error Type and ErrInfo Code	Error Location			
		Addr	Word	Way	Entry

0x7 (continued)	• [4] – Command Type (Reserved for CHI-AIU & DCE)
	  ○ 1'b0: Read
	  ○ 1'b1: Write
	• [5] – Reserved
	• [15:6] – Transaction ID/AXID (Reserved for DCE)
		Transaction Address			

0x8	Transport Error
	• [0] – Type
	  ○ 1'b0: Wrong Target Id
	  ○ 1'b1: Atomic Transaction w/o CCP
	• [5:1] – Reserved
	• [15:6] – Source ID (FUnit_Id)
		Transaction Address			

0x9	Timeout Error • [1:0] – Reserved • [2] - Security Attribute • [15:3] - Reserved	Transaction Address			

0xA	Sys Event Error
	• [0] – Type
	  ○ 1'b0: Time out error on message (protocol timeout)
	  ○ 1'b1: Time out error on interface
	• [15:1] – Reserved
		-			

0xB	SysCo Error
	• [0] – Type
	  ○ 1'b0: Time out error on message (protocol timeout)
	  ○ 1'b1: External error reported by response from system
	• [15:1] – Reserved
		-			

0xC – 0xF	Reserved	N/A	N/A	N/A	N/A

1.3 Native Interface Error Response Logging and Interrupt

Uncorrectable Error is logged when an error response is detected at the native interfaces if ProtErrDetEn = 1.

- DMI/DII logs all AXI Bresp/Rresp errors
- CHIAIU logs all CHI data/non-data response errors
- IOAIU logs all ACE CRresp errors (CRresp[1] is logged into ErrInfo Response field)

An Uncorrectable interrupt is raised if ProtErrIntEn = 1.

CCMP must be completed with error CmStatus for DtwReq and DtrReq, if the first beat of data contains error else the CmStatus does not indicate an error. If following beats of data contain error, it is indicated by setting the appropriate DBad of the data beat. Native interface to CmStatus mapping is shown in Table 1.

Native interface	Native interface error	CmStatus	Description
AXI (DMI/DII)	SLVERR	0b10000011	Data error in CmStatus. As per ARM spec this error is reported when the slave cannot provide data due to various reasons
	DECERR	0b10000100	Address error in CmStatus. As per ARM Spec this error is reported when the slave cannot be found by the network
CHI (CAIU)	Data Error	0b10000011	Data error in CmStatus. As per ARM spec this error is reported when data is corrupted. The error may be reported on datflit or rspflit.
	Non-Date Error	0b10000100	Address error in CmStatus. As per ARM Spec this error is reported when the response cannot be completed due to various reasons. The error may be reported on datflit or rspflit.
ACE (CAIU)	On CR channel bit 1	0b10000100	Address error in CmStatus. As per ARM spec, the agent cannot complete the snoop request. (Today RTL issues Target Signaled error for DVM snoops, this needs to be updated in later versions to address error) Concerto expects following: If the originating snoop was an invalidating type, then the agent is expected to go to invalid state else stay in the same state. Snoop filer is updated accordingly.

TABLE 1: NATIVE INTERFACE TO CMSTATUS MAPPING

In the cases where the data is provided at the native interface the error is propagated on either DtwReq or DtrReq. When data is not provided then the error is propagated on SnpRsp or DtwRsp.

In Ncore System, data can be rotated per DW (Double Word), the beat data error information from native interface is carried in each DW, and after rotation, certain error type can overwrite the others. For example, in AXI, if the first beat sent on SMI contains double word for both SLVERR and DECERR, Address Error is injected into CMStatus.

CmStatus to native interface mapping is shown in Table 2. In the case of DtwReq and DtrReq the first beat CmStatus may not indicate error, but consecutive beats may indicate DBad in this case the consecutive beats are treated as Data Error.

Native interface	CmStatus	Native interface error	Description
AXI/ACE-Lite/Ace-Lite E/ACE (NCAIU/CAIU)	0b10000011	SLVERR	Data error is best represented by SLVERR
	0b10000100	DECERR	Address error where the slave cannot be found is best represented by DECERR. In the case of DVMs (CmpRsp) this must be decoded to SLVERR
CHI (CAIU)	0b10000011	Data Error	Data error, this includes SLVERR (reported by DMI/DII) is best mapped to Data Error on CHI as Ncore does not have enough information to decode the SLVERR and it may be reported due to bad data
	0b10000100	Non-Data Error	Address error, this includes DECERR (reported by DMI/DII) is best mapped to Non-Data Error on CHI as this error is reported when the final target cannot be found.

TABLE 2 CMSTATUS TO NATIVE INTERFACE MAPPING

1.4 Transport Error Logging and Interrupt

Ncore 3 units support only Wrong Target ID error. Uncorrectable Error is logged when seeing message targetID (exclude Port_ID) mismatches unit's FUnit_ID if TransErrDetEn = 1. The message is dropped. An Uncorrectable interrupt is raised if TransErrIntEn = 1.

1.5 Resiliency Related Error Logging and Interrupt

SMI Protection Error:

- If both resiliency and SMI protection are enabled, it is FSC's responsibility to log and interrupt SMI protection error, which includes: Concerto Header Error, Concerto Message Error and Concerto Data Error.
- Upon an SMI protection Uncorrectable Error, the message is dropped by the unit and error is logged and interrupt by FSC.
- Upon a SMI Protection Correctable Error, the error is corrected by the unit and the error counter is incremented. If Error Count reaches ResThreshold and a new error is detected, the error is logged, and an interrupt is asserted by FSC.

Please see the resiliency section for FSC error handling.

In addition, following uncorrectable errors are reported to the fault checker in resiliency mode and will trigger mission fault:

- Uncorrectable memory errors which include both parity and ECC
- Wrong target ID error

1.6 Address Map Decode Error Logging and Interrupt

AIU and DCE detects and logs the following Uncorrectable Errors upon address decoding if DecDetErrEn = 1, If multiple of these errors are reported for the same transaction, then reporting order of priority is from top to bottom as below:

- No Address hit
  - If the target cannot be found in the address map
  - Or if AIUs want to issue CSR access when it is not allowed
- Multiple Address hit

        - If multiple targets are found in the address map
    - Illegal Security access
        - If Non-Secure transaction access secure region as configured in the address map
    - Illegal CSR access format
        - If the target is decoded as 32-bit CSR DII, and the following requirements are not met
            - 4 Byte
            - Size-aligned
            - EndPoint order
            - Device Transaction
    - Illegal DII access type
        - Coherent access on DII
    - Unconnected DMI unit access
        - If the target DMI as decoded by the address map is not connected to the AIU
    - Unconnected DII unit access
        - If the target DII as decoded by the address map is not connected to the AIU
    - Unconnected DCE unit access
        - If the target DCE as decoded by the address map is not connected to the AIU
    - No credits configured for access
        - If the target as decoded by the address map does not have any credits configured

An Uncorrectable interrupt is raised if DecErrIntEn = 1.

In addition, an AIU needs to complete the protocol and return error in the response.

- ACE/ACE-LITE/AXI: DECERR
- CHI: non-data error

A DCE needs to drop the recall transaction, i.e., pretend it finished successfully. In the case of 'no credits configured for access' error (Mrd Credits), DCE must issue an address error with in the CmStatus field of the StrReq.

1.7 CCMP Completion

AIU does not expect DtrReq if StrReq contains error CmStatus for a read request, and it needs to manufacture response to the processor.

When IOAIU detects an OTT data buffer uncorrectable error:

- DtwReq is returned with CMStatus = 8'b10000011 (Data Error) and poison bit set. If error is not detected on first beat, then only poison bit is set (writes)
- Data that is to be stored in cache the poison bit is set
- If the data is to be sent on to the native interface i.e., R Channel, RResp is returned with SLVERR (Reads)
- If the data is NOT in use by the original transaction, the error is ignored. i.e., original transactions of CleanUnique/MakeUnique, CMOs, DVMs, CacheStashing, etc.

When IOAIU is configurated with cache:

- For a tag Uncorrectable Error during cache lookup due to native interface request, IOAIU treats the request as a cache miss and furthermore does not allocate to the cache. In this case uncorrectable error is logged in CSrs and reported via interrupt.
- For a tag Uncorrectable Error during SNPReq lookup, IOAIU issues SNPRsp with CMStatus = 8'b10000100 (Address Error).
- For a data Uncorrectable Error upon access cache data RAM, if first beat data is poisoned and DtrReq/DtwReq is returned with CmStatus = 8'b10000011 (Data Error) and poison bit set, if a data beat after the first beat is poisoned then only poison bit is set and CmStatus may not indicate error. In the case where data is to be provided on the native interface i.e. R Channel, RResp is returned with SLVERR on the appropriate data beat.

When IOAIU processes a multi-line transaction:

- For read transaction each individual Cache line is for error reporting purposes is treated separately as if it was a single cache line transaction. Requirements are defined in the section above
- For write transaction the final response on the native interface must be the accumulation of all individual cache line write responses. Here Address error (DECERR) has higher precedence than (SLVERR).

When DCE gets a snoop filter look up error:

- DCE issues STRReq with CMStatus = 8'b10000100 (Address Error).

When DCE is configured with zero DMI credits:

- DCE issues STRReq with CMStatus = 8'b10000100 (Address Error).

When DCE gets any snoop response error:

- If any snooped AIU is providing the data, the protocol is completed by the DtrReq and the error is not propagated by DCE via StrReq. (Exception for read Stash below)
- If no snooped AIU is providing the data (DtrReq) then DCE issues StrReq with propagated error CmStatus from the last received SnpRsp with error. (Exception for read Stash below)
- In the case of Read Stash where the non-target reports SnpRsp error DCE does not propagate it and issues MrdReq to complete the request
- In the case of Read Stash where the target reports SnpRsp error, DCE propogates the error on StrReq and aborts the transaction
- In the case of Write stash and Write Unique SnpRsp errors are not propagated onto to StrReq

- If the original snoop was an invalidating snoop type (i.e the snooped agent is required to go to invalid state) then the snoop filter is updated to invalidate that agent, for all other snoop types, snoop filter status is not updated for that agent. (In general, it is expected that the agent that responded with an error went to an invalid state, but Ncore takes a more conservative approach due to ambiguity in ARM specs and only invalidates the snoop filter for cases where the snoop type required an invalidation)
  Invalidating snoop type are: *SnpInvDtw, SnpInvDtr, SnpInv, SnpInvStsh, SnpUnqStsh, SnpStshUnq*

When DMI is configurated with cache:

- For a tag Uncorrectable Error during SMC lookup due to CmdReq/ MrdReq/ DtwMrgMrdReq messages, DMI issues a control propagation DtrReq message that indicates an address corruption error. In addition, the DMI manufactures appropriately sized and poisoned data payload for the Dtrreq with CmStatus = 8'b10000100 (Address Error). Since the entire data payload is poisoned, the actual content of the payload is not important. For a tag error during SMC lookup due to a prefetch message, DMI drops the message. DMI completes the CCMP Prefetch transaction by sending STRreq without an error signal.
- For a data Uncorrectable Error upon access SMC data RAM, data is poisoned and DTRReq is returned with CMStatus = 8'b10000011 (Data Error)

When DVE gets a Snoop response error:

- DVE issues CmpRsp with CMStatus = 8'b10000100 (Address Error).

CMStatus error is propagated only on following responses:

- SnpRsp coming into DCE, only if no other related SnpRsp issued a DTR then this is propagated on to StrReq else it is not propagated on to StrReq
- DtwRsp coming into AIUs, this is propagated on to the native interface
- MrdRsp coming into DCE, this is propagated on to StrReq
- CmpRsp coming into AIUs, this is propagated on to the native interface

CMStatus error is propagated only on following requests:

- StrReq going out of DCE
- DtrReq going out of AIU, DMI, and DII
- DtwReq going out of AIU

In the case of coherent Atomic transactions, where there are two parts the coherent part (clean operation with DCE) and the non-coherent part (atomic transaction with DMI) following applies:

- If an error is reported back to the initiating AIU during the first coherent part, then the error is propagated on the Native interface and the second part is not carried out. Note that the AIU is expected to issue proper response and deallocate its transaction entry.
- If an error is reported back to the initiating AIU during the second non-coherent part, then the error is propagated on the Native interface. Note that the AIU is expected to issue proper response and deallocate its transaction entry.
- For Atomic Load/Swap/Compare where the BRESP and RDATA/RRESP is required, two responses can report different types of error from DTWrsp and DTRreq. An AIU needs to wait for both responses and return a consolidated identical response back to the Native interface.
- The priority of error types is: Address Error (DECERR) > Data Error (SLVERR) > OK

In the case of ACE and IOAIU with proxy cache configurations where WriteBacks, WriteClean, WriteEvict, Evict and cache eviction can have two parts i.e. non-coherent write to DMI and coherent update to DCE. If an error is reported back during

the non-coherent transaction part, then the coherent update parts needs to be completed. In the case of ACE the error needs to be propagated on the native interface.

1.8 Data Poisoning

When seeing any DBad on DtwReq, the receiver should not log error and just issue normal CMStatus in the response. However, the data is marked as poisoned if filling into a cache and the write strobes for the whole beat are de-asserted if writing to downstream.
Note that there can be cases where the CMStatus of the DtrReq and DtwReq may not report the error but DBad may be set. This happens in cases where the error was detected after processing the first beat of data.
In DMI, if any error is detected in CMStatus, the poison bits of whole cache line will be set and propagate on DTRreq.
In IOAIU with proxy cache, partial write data will be merged with DtrReq to generate the fill data to CCP, the poison bits are in per CCP data beat based (usually CCP data bank width = AXI data width = DTR data beat width), if BEs from AXI write data of entire CCP beat are set, the poison bit of that beat will be overwritten with poison bit from AXI write data.

1.9 Timeout Error

Timeout counters are implemented by AIUs, DCEs and DMIs; latter two implement it specifically to handle timeout due to recall or evictions transactions respectively. Every other transaction is covered by AIUs.

Timeout Detection:

One global counter is designed to count cycles. Once the counter reaches a programmable threshold, the timeout overflow bit in all valid entries is set. This indicates the start of the period where the entry must make progress. The timeout overflow bit will be cleared if the entry deallocates. If the global counter reaches the threshold again while the timeout overflow bit is still set, then that entry has timed out. The event is logged in an error register to allow an interrupt to be raised if enabled.

Timeout Handling:

The status is logged in the CSR software will now have the option to either disable timeout and continue or treat this as a fatal error. The timeout counter will stop running, until software restarts it via CSR.

1.10 Sys event and SysCo Errors

Sys event sender can report a timeout error if it does not get a SysRsp message for the SysReq it sent out. This is also referred to as protocol timeout.

Sys event receiver can report a timeout error if it does not get an ack back on the event pins.

Sys Co handler can report a timeout error if it does not get a SysRsp message for the SysReq it sent out.

1.11 Unsupported Message Type

If a Ncore unit received an unsupported Message (for example DMI gets a snoop command), the message will not be processed and eventually trigger a timeout error.

1.12 Error Reporting in CMStatus

If multiple errors are detected in a message, NCore Unit will report control errors (eg. Address Error) over data error in CMStatus.

2 Opens

Questions/Feedback/Need to discuss:

3 Glossary

Arteris

A NoC Company

NCore3

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

4 Notes

Notes ……