



AMBA CHI Issue E.b Errata

Architecture & Technology Group

Document number: ARM AES 0061
Release Number: 2.0
Confidentiality: Non-Confidential
Date of Issue: 25-Apr-2022

© Copyright Arm Limited 2022. All rights reserved.

Abstract

This document includes clarifications and corrections to the CHI Issue E.b specification.

Contents

About this document		3
Release Information		3
References		3
1	Introduction	7
1.1	Classification of the change	7
2	Errata list	8
2.1	C605: DBID values in Comp and DBIDResp messages that originate from difference sources in DWT flow have no relationship	8
2.2	C606: Data_Check and Check_Type property descriptions	9
2.3	D607: Typographical error in Allocating Read figure	10
2.4	D609: Typographical error in RN to HN Write request attribute values table	11
2.5	C612: Requirements for completer read response when MTE is not supported	12
2.6	C621: Typographical error in example WriteUniqueStash with Data Pull transaction flow	13
2.7	C624: DVM payload encoding for Instruction cache invalidations	14
2.8	C628: Requirements for ReadOnceMakeInvalid when invalidating a Dirty copy	15

About this document

Release Information

The change history table lists the changes that have been made to this document.

Date	Version	Confidentiality	Change
03-Feb-2022	1.0	Confidential	First limited release
25-Apr-2022	2.0	Non-Confidential	First public release

References

This document refers to the following documents.

Ref	Document Number	Title
1	ARM IHI 0050E.b	AMBA® 5 CHI Architecture Specification

Proprietary Notice

Proprietary Notice

This document is **NON-CONFIDENTIAL** and any use by you is subject to the terms of this notice and the Arm AMBA Specification Licence set out below.

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

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

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

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its 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 Arm's trademark usage guidelines at <http://www.arm.com/company/policies/trademarks>.

Copyright © [2022] Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-21451 version 2.2

AMBA SPECIFICATION LICENCE

THIS END USER LICENCE AGREEMENT ("LICENCE") IS A LEGAL AGREEMENT BETWEEN YOU (EITHER A SINGLE INDIVIDUAL, OR SINGLE LEGAL ENTITY) AND ARM LIMITED ("ARM") FOR THE USE OF ARM'S INTELLECTUAL PROPERTY (INCLUDING, WITHOUT LIMITATION, ANY COPYRIGHT) IN THE RELEVANT AMBA SPECIFICATION ACCOMPANYING THIS LICENCE. ARM LICENSES THE RELEVANT AMBA SPECIFICATION TO YOU ON CONDITION THAT YOU ACCEPT ALL OF THE TERMS IN THIS LICENCE. BY CLICKING "I AGREE" OR OTHERWISE USING OR COPYING THE RELEVANT AMBA SPECIFICATION YOU INDICATE THAT YOU AGREE TO BE BOUND BY ALL THE TERMS OF THIS LICENCE.

"LICENSEE" means You and your Subsidiaries.

"Subsidiary" means, if You are a single entity, any company the majority of whose voting shares is now or hereafter owned or controlled, directly or indirectly, by You. A company shall be a Subsidiary only for the period during which such control exists.

1. Subject to the provisions of Clauses 2, 3 and 4, Arm hereby grants to LICENSEE a perpetual, non-exclusive, non-transferable, royalty free, worldwide licence to:

(i) use and copy the relevant AMBA Specification for the purpose of developing and having developed products that comply with the relevant AMBA Specification;

(ii) manufacture and have manufactured products which either: (a) have been created by or for LICENSEE under the licence granted in Clause 1(i); or (b) incorporate a product(s) which has been created by a third party(s) under a licence granted by Arm in Clause 1(i) of such third party's AMBA Specification Licence; and

(iii) offer to sell, sell, supply or otherwise distribute products which have either been (a) created by or for LICENSEE under the licence granted in Clause 1(i); or (b) manufactured by or for LICENSEE under the licence granted in Clause 1(ii).

2. LICENSEE hereby agrees that the licence granted in Clause 1 is subject to the following restrictions:

(i) where a product created under Clause 1(i) is an integrated circuit which includes a CPU then either: (a) such CPU shall only be manufactured under licence from Arm; or (b) such CPU is neither substantially compliant with nor marketed as being compliant with the Arm instruction sets licensed by Arm from time to time;

(ii) the licences granted in Clause 1(iii) shall not extend to any portion or function of a product that is not itself compliant with part of the relevant AMBA Specification; and

(iii) no right is granted to LICENSEE to sublicense the rights granted to LICENSEE under this Agreement.

3. Except as specifically licensed in accordance with Clause 1, LICENSEE acquires no right, title or interest in any Arm technology or any intellectual property embodied therein. In no event shall the licences granted in accordance with Clause 1 be construed as granting LICENSEE, expressly or by implication, estoppel or otherwise, a licence to use any Arm technology except the relevant AMBA Specification.

4. THE RELEVANT AMBA SPECIFICATION IS PROVIDED "AS IS" WITH NO REPRESENTATION OR WARRANTIES EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF SATISFACTORY QUALITY, MERCHANTABILITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE, OR THAT ANY USE OR IMPLEMENTATION OF SUCH ARM TECHNOLOGY WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADE SECRETS OR OTHER INTELLECTUAL PROPERTY RIGHTS.

5. NOTWITHSTANDING ANYTHING TO THE CONTRARY CONTAINED IN THIS AGREEMENT, TO THE FULLEST EXTENT PERMITTED BY LAW, THE MAXIMUM LIABILITY OF ARM IN AGGREGATE FOR ALL CLAIMS MADE AGAINST ARM, IN CONTRACT, TORT OR OTHERWISE, IN CONNECTION WITH THE SUBJECT MATTER OF THIS AGREEMENT (INCLUDING WITHOUT LIMITATION (I) LICENSEE'S USE OF THE ARM TECHNOLOGY; AND (II) THE IMPLEMENTATION OF THE ARM TECHNOLOGY IN ANY PRODUCT CREATED BY LICENSEE UNDER THIS AGREEMENT) SHALL NOT EXCEED THE FEES PAID (IF ANY) BY LICENSEE TO ARM UNDER THIS AGREEMENT. THE EXISTENCE OF MORE THAN ONE CLAIM OR SUIT WILL NOT ENLARGE OR EXTEND THE LIMIT. LICENSEE RELEASES ARM FROM ALL OBLIGATIONS, LIABILITY, CLAIMS OR DEMANDS IN EXCESS OF THIS LIMITATION.

6. No licence, express, implied or otherwise, is granted to LICENSEE, under the provisions of Clause 1, to use the Arm tradename, or AMBA trademark in connection with the relevant AMBA Specification or any products based thereon. Nothing in

Clause 1 shall be construed as authority for LICENSEE to make any representations on behalf of Arm in respect of the relevant AMBA Specification.

7. This Licence shall remain in force until terminated by you or by Arm. Without prejudice to any of its other rights if LICENSEE is in breach of any of the terms and conditions of this Licence then Arm may terminate this Licence immediately upon giving written notice to You. You may terminate this Licence at any time. Upon expiry or termination of this Licence by You or by Arm LICENSEE shall stop using the relevant AMBA Specification and destroy all copies of the relevant AMBA Specification in your possession together with all documentation and related materials. Upon expiry or termination of this Licence, the provisions of clauses 6 and 7 shall survive.

8. The validity, construction and performance of this Agreement shall be governed by English Law.

Confidentiality Status

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

Product Status

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

Web Address

<http://www.arm.com>

1 Introduction

This document lists errata on AMBA CHI Issue E.b.

Each errata description is organized as a brief reason for the change, along with the precise change.

1.1 Classification of the change

Each listed item has a classification ID, of the form XYYY, where:

X is the errata classification type as follows, C, R, E or D:

C	Clarification	Informative change only
R	Relaxation	Backward-compatible normative change, modifying existing functionality
E	Enhancement	Backward-compatible normative change, adding new functionality
D	Defect	Non-backward compatible normative change

YYY is an Arm internal tracking number.

2 Errata list

2.1 C605: DBID values in Comp and DBIDResp messages that originate from difference sources in DWT flow have no relationship

Affects:

CHI-E.a, CHI-E.b

Description:

When a write transaction takes advantage of the DWT flow, the source of the DBIDResp and Comp response messages into the original requester will be different. DBIDResp will originate from the Subordinate and Comp will originate from the Home.

There is no requirement or relationship between the Comp.DBID value returned by the Home and the DBIDResp.DBID value returned by the Subordinate during DWT.

The precise change(s):

On page 2-90, in section 2.5.9 "Data Buffer ID" the following text:

A Comp response message sent separate from a DBIDResp or DBIDRespOrd message for a Write transaction must include the same DBID field value in the Comp and DBIDResp or DBIDRespOrd message.

Will be updated to:

A Comp response message sent separate from a DBIDResp or DBIDRespOrd message for a Write transaction must include the same DBID field value in the Comp and DBIDResp or DBIDRespOrd message when the two messages originate from the same source.

In addition, at the bottom page 2-106 in the "WriteNoSnp transaction" identifier field usage description, the following Note:

There is no ordering requirement between the separate DBIDResp and Comp responses. It is required that the values used are identical.

Will be updated to:

There is no ordering requirement between the separate DBIDResp and Comp responses. It is required that the values used are identical when the two messages originate from the same source.

2.2 C606: Data_Check and Check_Type property descriptions

Affects:

CHI-D, CHI-E.a, CHI-E.b

Description:

The Data_Check and Check_Type properties are used to declare the parity capabilities supported on an interface. There is an overlap between these two properties and additional clarity will be added to the specification to cover the case where they could potentially be viewed as being in contention. The parity support on an interface will be one of the following:

- No checking
- DataCheck field in the DAT packet
- Complete interface parity checking

The precise change(s):

On page 9-348 in section 9.6 "Data Check" the following line will change from:

The Data_Check property is used to indicate if Data Check is supported.

To:

The Data_Check and Check_Type properties are used to indicate if Data Check is supported in the DAT packet.

On pages 16-470 and 16-471 the Data_Check and Check_Type property descriptions in 16.1 "Interface properties and parameters" will be replaced with:

Data_Check

The Data Check property is used to indicate if the DataCheck field is present in the DAT packet.

- When not specified, or set to *False*, the DataCheck field is not present in the DAT packet unless specified by the Check_Type property.
- When set to *Odd_Parity*, Data Check is supported and the DataCheck field is present in the DAT packet. If Check_Type is defined and set to *Odd_Parity_Byte_All*, no DataCheck field is present in the DAT packet.

Check_Type

The Check Type property is used to indicate the protection scheme employed on an interface:

When not specified or set to *False*, no checking signals are present on the interface, unless specified by the Data_Check property.

- When set to *Odd_Parity_Byte_Data*, the DataCheck field is present in the DAT packet.
- When set to *Odd_Parity_Byte_All*, parity check signals are added to every channel. The signals added are detailed in 9.7.3 Interface parity check signals on page 9-351.

2.3 D607: Typographical error in Allocating Read figure

Affects:

CHI-E.b

Description:

The figure detailing the possible transaction flows for an Allocating Read Transaction is updated to correct a typographical error.

The precise change(s):

On page 2-42, Figure 2-1 "Allocating Read" will have SnpRespFwdwd replaced with SnpRespFwded

2.4 D609: Typographical error in RN to HN Write request attribute values table

Affects:

CHI-E.b

Description:

Table 4-13 lists the values permitted for key attributes in Write requests from Request Nodes to Home Nodes. It incorrectly lists WriteBackPtl twice, the second of which is in error and should be WriteCleanFull.

The precise change(s):

On page 4-178 in Table 4-13 "RN to HN Write request attribute values" replace:

WriteBackFull

WriteBackPtl

with:

WriteBackFull

WriteCleanFull

2.5 C612: Requirements for completer read response when MTE is not supported

Affects:

CHI-E.a, CHI-E.b

Description:

The Memory Tagging Extension (MTE) is a mechanism that is used to check the correct usage of data held in memory. In a given system, not all devices/addresses may support MTE. When a Read request is issued that also asks for MTE Tags (TagOp value of *Transfer* or *Fetch*), if the end device does not support MTE, the response must use a TagOp value of *Invalid*.

Section 12.11.3 "MTE not supported" can be interpreted as relaxing this constraint with its use of the term permitted, suggesting that there might be an alternative response that could also be given when this is not true.

The precise change(s):

On page 12-387 in section 12.11.3 "MTE not supported" the following text will change from:

When a Completer does not support MTE for the address in the request, then the Completer is permitted to send a TagOp value of *Invalid* in response to a Read transaction with a TagOp value of *Transfer* or *Fetch*.

To:

When a Completer does not support MTE for the address in the request, then the Completer must send a TagOp value of *Invalid* in response to a Read transaction with a TagOp value of *Transfer* or *Fetch*.

2.6 C621: Typographical error in example WriteUniqueStash with Data Pull transaction flow

Affects:

CHI-B, CHI-C, CHI-D, CHI-E.a, CHI-E.b

Description:

Figure 5-23 shows an example WriteUniqueStash with Data Pull transaction flow. The SnpResp provided by RN-F1 includes a Data Pull request, asking for the associated line to be returned to it. The HN-F returns CompData in a UniqueDirty state with PassDirty indicated. This means that RN-F1 must allocate the line in that state, but a label indicating this on the figure is currently missing.

The precise change(s):

At the arrow head signaling the return of CompData_UD_PD to RN-F1, a label of "I->UD" will be added to Figure 5-23.

2.7 C624: DVM payload encoding for Instruction cache invalidations

Affects:

CHI-E.a, CHI-E.b

Description:

To help clarify the field positioning in the DVMOp and SnpDVMOp request payloads, CHI-E.b had a new implementation of Table 8-8 vs CHI-E.a. The table wrongfully omitted the field locations of VI Valid and Virtual Index VA. The additional tables that detailed the various DVM operations also had their bit position values removed, as these differ between the DVMOp and SnpDVMOp request payloads due to the width of the Req.Addr and Snp.Addr fields. This combination of changes could cause confusion when establishing the encoding for Physical Instruction Cache Invalidate operations.

The precise change(s):

Table 8-8 will be modified in the following ways:

- Adding of "VI Valid[0]" to Req.Addr[5]/Dat.Data[5] row in the Request column where "VMID Valid" currently exists
- Adding of "VI Valid[0]" to Snp.Addr[2] row in the Part 1 column where "VMID Valid" currently exists
- Adding of "VI Valid[1]" to Req.Addr[6]/Dat.Data[6] row in the Request column where "ASID Valid" currently exists
- Adding of "VI Valid[1]" to Snp.Addr[3] row in the Part 1 column where "ASID Valid" currently exists
- Adding of "Virtual Index VA[27:20]" to Req.Addr[21:14]/Dat.Data[21:14] row in the Request column where "VMID[7:0]" currently exists
- Adding of "Virtual Index VA[27:20]" to Snp.Addr[18:11] row in the Request column where "VMID[7:0]" currently exists
- Adding of "Virtual Index VA[19:12]^d" to Req.Addr[37:22]/Dat.Data[37:22] row in the Request column where "ASID[15:0]" currently exists
- Adding of "Virtual Index VA[19:12]^d" to Snp.Addr[34:19] row in the Request column where "ASID[15:0]" currently exists

Additionally, the following footnote will be added to Table 8-8:

- d. When used as Virtual Index VA, Req.Addr[37:30], Dat.Data[37:30] and Snp.Addr[34:27] can take any value

Table 8-26 will have the column header "Virtual Index" changed to "VI Valid".

2.8 C628: Requirements for ReadOnceMakeInvalid when invalidating a Dirty copy

Affects:

CHI-B, CHI-C, CHI-D, CHI-E.a, CHI-E.b

Description:

The completion of a ReadOnceMakeInvalid may result in a Dirty copy in the system being invalidated. This will be dependent on whether or not the Home accepts the invalidation hint included as part of the request. In order to ensure that the system remains coherent, if a Dirty copy is invalidated and not written back to memory, all other cached copies must be invalidated.

The precise change(s):

In section 4.2.1 on page 164 of CHI-E.b, the following paragraph will be updated from:

Read request to a Snoopable address region to obtain a snapshot of the coherent data. It is recommended, but not required, that all snoopable cached copies are invalidated. If a Dirty copy is invalidated, it does not need to be written back to memory.

To:

Read request to a Snoopable address region to obtain a snapshot of the coherent data. If a Dirty copy is invalidated, it does not need to be written back to memory. All cached copies must be invalidated if the invalidation hint is accepted and a Dirty copy is not written back to memory.