AMBA® CHI Issue G Errata

Architecture & Technology Group

Document number	ARM-AES-0111
Document quality	Released
Document version	4.0
Document confidentiality	Non-confidential
Date of issue	07-Feb-2025

AMBA® CHI Issue G Errata

Release information

Date	Version	Changes
2025/Feb/07	4.0	• Second public release
2024/Dec/18	3.0	• Second limited release
2024/Oct/21	2.0	• First public release
2024/Sep/27	1.0	• First limited release

Non-Confidential Proprietary Notice

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-21451 version 3

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 PETMITTED 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.

PRE-21451 version 3

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

Contents

AMBA® CHI Issue G Errata

Preface

This document lists errata on the AMBA CHI Issue G specification [1].

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

Errata classification

Each listed errata 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.

Additional reading

This section lists publications by Arm and by third parties.

See Arm Developer http://developer.arm.com for access to Arm documentation.

[1] *AMBA 5 CHI Architecture Specification*. (ARM IHI 0050 G) Arm Ltd.

Chapter 1
New/Updated Errata

1.1 D1017: RespSepData with non-Invalid Resp considered to be globally observed

Affects:

CHI-C, CHI-D, CHI-E.a, CHI-E.b, CHI-E.c, CHI-F, CHI-F.b, CHI-G

Description:

CHI-C introduced the option of separate response, RespSepData, and data, DataSepResp, for read transactions. This was a performance improvement that enabled for a decrease in transaction lifetime at the Home Node when compared against the earlier CompData only approach.

When originally specified, the ordering semantics of the two responses were:

- RespSepData
  - The relevant request has been ordered at Home.
  - The Requester does not receive any snoops for transactions that are scheduled before the RespSepData response.
    - Home must ensure that no Snoop transactions are outstanding to that Requester to the same address.
  - Receiving of RespSepData does not guarantee that Home has completed snooping of other agents in the system.
- DataSepResp
  - The Read transaction can be considered to be globally observed.

The Resp field value in RespSepData is either:

- Inapplicable and set to 0 (Invalid).
- Set to the same value as in the corresponding DataSepResp.

For a Home Node to provide a non-Invalid Resp field value, it is committing that DataSepResp response will have the same Resp field value. For this confidence level to be in place, it is highly likely that any coherence action associated with the request has already been completed.

Arm aims to retrospectively tighten the ordering semantics specifically for the cases when RespSepData is provided with a non-Invalid Resp field value, such that this response also provides a global observably guarantee. This guarantee permits Processing Elements (PEs) to clear related ordering hazards, improving performance.

There are no changes being made to the the snoop hazard conditions that an RN-F must adhere to in relation to the RespSepData and DataSepResp responses. That is, if an RN-F has received RespSepData response, the RN-F must wait to receive all DataSepResp response packets before responding to a Snoop request targeting the same address.

The precise change(s):

In section B2.6.4 “Ordering semantics of RespSepData and DataSepResp” of CHI-G on page 132, the following text will be updated from:

> When a Requester receives the first DataSepResp, the Read transaction can be considered to be globally observed. This is because there is no action which can modify the read data received.
>
> When a Requester receives a RespSepData response from Home, the relevent request has been ordered at Home. The Requester does not receive any snoops for transactions that are scheduled before the RespSepData response. Before sending RespSepData response to the Requester, the Home must ensure that no Snoop transactions are outstanding to that Requester to the same address. Receiving of RespSepData does not guarantee that Home has completed snooping of other agents in the system.

To:

When a Requester receives the first DataSepResp, the Read transaction can be considered to be globally observed. This is because there is no action which can modify the read data received.

When a Requester receives a RespSepData response from Home, the relevant request has been ordered at Home. The Requester does not receive any snoops for transactions that are scheduled before the RespSepData response. Before sending RespSepData response to the Requester, the Home must ensure that no Snoop transactions are outstanding to that Requester to the same address.

When a Requester receives:

– RespSepData with a Resp field value of Invalid, the Read transaction cannot be considered to be globally observed. That is RespSepData_I does not guarantee that Home has completed snooping of other agents in the system.

– RespSepData with any legal Resp field value other than Invalid, the Read transaction can be considered to be globally observed.

Additionally, the in Table B2.7 “Completion response and ordering” on page 128, the Read major row will be updated from:

Transaction	Location	Response	Outcome
Read	Cacheable	CompData or DataSepResp	The transaction is observable to a later transaction from any agent to the same location.
	Cacheable	RespSepData	No earlier transaction will send a snoop to this Requester. All later transactions send a snoop only if required after the Home receives the CompAck response for this transaction.
	Non-cacheable or Device	RespSepData or CompData	The transaction is observable to a later transaction from any agent to the same endpoint address range.

To:

Transaction	Location	Response	Outcome
Read	Cacheable	CompData or DataSepResp or RespSepData_UC or RespSepData_SC or RespSepData_UD_PD or RespSepData_SD_PD	The transaction is observable to a later transaction from any agent to the same location.
	Cacheable	RespSepData_I	No earlier transaction will send a snoop to this Requester. All later transactions send a snoop only if required after the Home receives the CompAck response for this transaction.
	Non-cacheable or Device	RespSepData or CompData	The transaction is observable to a later transaction from any agent to the same endpoint address range.

1.2 R1021: StashOnce* permitted to drive non-zero DataTarget.CacheLevel value

Affects:

CHI-G

Description:

The CHI-G release introduced a new subfield for DataTarget called CacheLevel. This hint can be used alongside CopyBack requests to help with the intentional placement of data within a system.

The main motivation for this subfield was the PRFM SLC feature added to the Arm Architecture. The PRFM SLC operation can be executed by a CPU regardless of whether the data is present in the local cache. However, it was initially expected to be most beneficial in a producer-consumer model where the Requester had updated the data locally.

Further performance analysis has indicated that allowing Requesters to use the CacheLevel subfield in StashOnce* transactions, in addition to CopyBacks, would be advantageous. Therefore, the specification will be adjusted to permit StashOnce* Opcodes to also utilize non-zero values on CacheLevel.

Arm does not anticipate any implementation issues, as the CacheLevel subfield is merely a hint signal that can be ignored.

The precise change(s):

The following text in section B11.3.3 Subfield applicability of CHI-G, on page 406, will be updated from:

> Although the Replacement and UnusedPrefetch subfields are most useful in CopyBack requests, their use is not restricted to CopyBack transactions. Use of the Replacement and UnusedPrefetch subfields is extended to all requests from the Request Node to HN-F, except for the following:
>
> – Atomics
>
> – Stash transactions when StashNIDValid is 1
>
> – PrefetchTgt
>
> – PCrdReturn
>
> – DVMOp
>
> The CacheLevel subfield can take any value in the following CopyBack Write transactions:
>
> – WriteBackPtl
>
> – WriteBackFull
>
> – WriteCleanFull
>
> – WriteEvictFull
>
> – WriteEvictOrEvict
>
> The CacheLevel subfield must be set to 0 in all other requests where the DataTarget is applicable.

To:

> The Replacement and UnusedPrefetch subfields can take any value in in all requests from the Request Node to HN-F, except for the following:
>
> – Atomics
>
> – Stash transactions when StashNIDValid is 1
>
> – PrefetchTgt

- PCrdReturn
- DVMOp

The CacheLevel subfield can take any value in the following requests:

- StashOnceUnique
- StashOnceShared
- StashOnceSepUnique
- StashOnceSepShared
- WriteBackPtl
- WriteBackFull
- WriteCleanFull
- WriteEvictFull
- WriteEvictOrEvict

The CacheLevel subfield must be set to 0 in all other requests where the DataTarget is applicable.

Chapter 2
Previous Errata

2.1 R858: BROADCASTMTE effect on MTE fields relating to previously cached locations

Affects:

CHI-E.a, CHI-E.b, CHI-E.c, CHI-F, CHI-F.b, CHI-G

Description:

When the MTE feature was introduced into CHI-E, a **BROADCASTMTE** pin was also included to control the MTE-related fields in messages. The intent of this signal is to ensure that Tag-related operations are not issued to downstream components that do not support MTE. It is expected that this pin will control the main functional logic of the component, but not necessarily the caches that exist towards the external interface of that component.

The MTE feature allows a request with TagOp=*Invalid* to see the corresponding data returned with TagOp=*Transfer*, indicating that Clean tags are included.

If a component that supports MTE has the **BROADCASTMTE** pin deasserted, it is possible for that component to cache Clean tags returned to a read request issued with TagOp=*Invalid*. When the data for that location is later sent downstream as a result of an eviction, or returned as a result of a snoop, it is possible for the data to be transferred along with the Clean tags. This should not cause a problem for the interconnect, as to receive the Clean tags in the first instance, it must already support MTE. It is always OK for a Requester to ignore or drop Clean tags that are returned to it.

Implementations are not expected to zero the MTE related fields when data allocated into the local cache is later transferred elsewhere. The previous wording for the **BROADCASTMTE** pin is considered too strict, and will be updated.

The precise change(s):

The text that currently describes **BROADCASTMTE**, in section B16.2.6 of CHI-G on page 537 will be updated.

From:

> The **BROADCASTMTE** signal is used to control the issuing of requests with MTE beyond the interface:
>
> – When asserted, requests with MTE can be sent beyond the interface.
>
> – When deasserted, requests with MTE must not be sent beyond the interface.
>
> The **BROADCASTMTE** signal is typically deasserted when the interface does not support MTE functionality.
>
> When the **BROADCASTMTE** signal is deasserted, all other MTE-related interface pins must be tied to 0. The interconnect is permitted, but not required, to remove the related MTE transport wires.
>
> The interface fields that can be fixed to a value of 0 are:
>
> – On DAT channels:
>
> – TagOp, Tag, and TU
>
> – On REQ and RSP channels:
>
> – TagOp

To:

The **BROADCASTMTE** signal is used to control the value of MTE related fields in messages beyond the interface except for those relating to evictions or snoop responses for cached data.

- When asserted, all messages with MTE can be sent beyond the interface.
- When deasserted:
  - In the following outbound REQ and DAT messages from a Requester Node, TagOp is permitted to be *Invalid* or *Transfer*:
    - Cache evictions, and any associated data transfers
    - SnpRespData, SnpRespDataFwded, and CompData in response to a snoop
  - In outbound DAT messages:
    - When TagOp is *Transfer*, Tag is permitted to be non-zero.
    - When TagOp is *Invalid*, Tag must be zero.
    - The TU field must be zero.
  - TagOp must be *Invalid* in:
    - All other outbound messages from a Requester Node.
    - All outbound messages from a Home Node.

Note
Cache evictions can be:

- Snoopable cache evictions for locations cached following Allocating Reads:
  - CopyBack Write
  - Combined CopyBack Write and CMO
- Non-snoopable cache evictions for locations cached following ReadNoSnp transactions:
  - WriteNoSnp
  - Combined WriteNoSnp and CMO

When **BROADCASTMTE** is deasserted it remains possible for cache evictions and snoop responses to be issued with MTE-related interface fields not set to 0. This can occur when a read issued with TagOp=*Invalid* sees the corresponding data returned with TagOp=*Transfer*, indicating Clean tags are included. If those Clean tags are cached, an implementation is expected, but not required, to zero out them out again when issuing a later CopyBack/WriteNoSnp of the line, or responding to a snoop that targets the cached line.

For the Requester to receive MTE tags, the interconnect must already be supporting MTE, so the later issuing of these tags will not cause any system issues.

The **BROADCASTMTE** signal is typically deasserted when the connecting interface does not support MTE functionality.

2.2 R907: When MEC_Support is False, MECID_Width can be non-zero

Affects:

CHI-G

Description:

When the *Memory Encryption Contexts* (MEC) feature was introduced, two additional properties were defined:

- MEC_Support
- MECID_Width

Rules were included that stated if MECID_Width was non-zero, then the MEC_Support property must be true. This is too restrictive as implementations might want to build in reset time configuration options that allow for the disabling of a feature property, whilst still keeping the associated fields present.

The rules will be relaxed so that the MECID_Width property being non-zero does not mandate the functionality associated MEC_Support.

For the MEC_Support feature property to be true, MECID field presence will still remain a requirement.

The precise change(s):

The following text on page 533 of the CHI-G specification will be updated from:

> The MECID_Width parameter has the following conditions:
>
> - If MEC_Support is False, MECID_Width must be 0
> - If MEC_Support is True, MECID_Width must not be 0.

To:

> The MECID_Width parameter has the following conditions:
>
> - If MEC_Support is True, MECID_Width must not be 0.
> - If MEC_Support is False, MECID_Width can take any permitted value.

Additionally, the “False or Not specified” row in Table B16.25 (MEC_Support property options) on page 532 will be updated to remove the following statement:

> No MECID fields are present on the interface.

2.3 C925: Resp can be non-Invalid in response to StashOnce*

Affects:

CHI-B, CHI-C, CHI-D, CHI-E.a, CHI-E.b, CHI-E.c, CHI-F, CHI-F.b, CHI-G

Description:

Cache stashing is a feature that was first introduced into CHI in issue B. The Completion response, Comp, to an independent Stash request is permitted to have a Resp value other than 0b000 (Invalid). The Resp value in the Comp response can be used to indicate the state of the cache line at the next level. This information might be useful for the tuning of HW prefetchers for example. If the prefetcher establishes that many of the previous stashes to the SLC were already present there prior to the StashOnce transactions being issued, then it could be considered that future StashOnce transactions to other locations might not be needed.

There have been contradictory statements in the CHI specifications to this point, some which permitted a non-zero Resp value in Comp for independent Stash transactions, and others which stated it must be 0b000 (Invalid).

Updates will be made to permit non-zero Resp values in Comp responses for independent Stash transactions.

The precise change(s):

The following text in section B4.5.1.2 (Dataless transaction completion) of CHI-G, on page 225, will be updated from:

Cache state The final state the cache line is permitted to be in at the Requester, except for CMO transactions. For CMO transactions, the cache state field value in the completion, specifically in Comp, CompCMO and CompPersist transactions, is ignored and the cache state remains unchanged.

To:

Cache state The final state the cache line is permitted to be in at the Requester, except for CMO transactions and StashOnce* transactions.

- For CMO transactions, the cache state field value in the completion, specifically in Comp, CompCMO and CompPersist responses, is ignored and the cache state remains unchanged.
- For StashOnce* transactions, the cache sate field value in the Comp and CompStashDone responses is permitted, but not required, to be used to indicate the presence of the cache line at the next level. See section B7.3 for more information.

Table B4.27 (Permitted Dataless transaction completion and Resp field encodings) will be updated to permit Comp_SD from the next level cache in response to a StashOnce transaction.

Table 2.1: Permitted Dataless transaction completion and Resp field encodings

Response	Resp[2:0]	Final cache line state	Notes
Comp_I	0b000	I	
Comp_UC	0b010	UD, UC, UCE, SC, or I	
Comp_SC	0b001	SC or I	
Comp_SD	0b011	N/A	Only used in Comp or CompStashDone for StashOnce transactions.
Comp_UD_PD	0b110	UD or SD	Responsibility for a Dirty cache line is being passed.

Table B13.35 (Valid Resp value encodings for different message types) will have the following text changed from:

Comp responses (Not including WriteNoSnpDef transactions)

To:

Comp responses (Not including WriteNoSnpDef or StashOnce* transactions)

A new major row will then be added to cover the permitted Resp values for StashOnce Comp responses:

Table 2.2: Valid Resp value encodings for different message types

Response type	Resp[2:0]	State	Notes
Comp responses for StashOnce* only	0b000	I	Cache state in the response is imprecise and must be ignored
	0b001	SC	Cache state in the next level cache
	0b010	UC, UD	
	0b011	SD	
	0b100	-	Reserved
	0b101	-	
	0b110	-	
	0b111	-	

2.4 D926: Some Table B10.4 encodings for PAS are incorrect

Affects:

CHI-G

Description:

CHI-G introduced the RME *Coherent Device Assignment* (CDA) feature, which potentially requires the SecSID1 field in device requests to the host. The SecSID1 field qualifies the Security state of the StreamID, which is then used in the translation process to determine the *Physical Address Space* (PAS) that can be accessed.

The table outlining the allowed relationship between SecSID1 and the final PAS mapping contains incorrect encodings for NSE and NS for the specified PAS. These errors will be corrected.

The precise change(s):

Table B10.4 (Field value relationships in a device request to the host) on page 394 of CHI-G will be updated from:

SecSID1	Security state	NSE	NS	PAS	Permitted combination
0b0	Non-Secure	0b0	0b1	Non-secure	Yes
		0b1	0b1	Realm	No
0b1	Realm	0b1	0b0	Non-secure	Yes
		0b0	0b0	Realm	Yes
0bX	Any	0b1	0b1	Root	No
		0b0	0b1	Secure	No

To:

SecSID1	Security state	NSE	NS	PAS	Permitted combination
0b0	Non-Secure	0b0	0b0	Secure	No
		0b0	0b1	Non-secure	Yes
		0b1	0b0	Root	No
		0b1	0b1	Realm	No
0b1	Realm	0b0	0b0	Secure	No
		0b0	0b1	Non-secure	Yes
		0b1	0b0	Root	No
		0b1	0b1	Realm	Yes