Errata for the Compute Express Link Specification Revision 3.1

December 13, 2023

LEGAL NOTICE FOR THIS SPECIFICATION FROM COMPUTE EXPRESS LINK CONSORTIUM, INC.

This CXL Errata for the Compute Express Link Specification (this "CXL Specification" or this "document") is owned by and is proprietary to Compute Express Link Consortium, Inc., a Delaware nonprofit corporation (sometimes referred to as "CXL" or the "CXL Consortium" or the "Company") and/or its successors and assigns.

NOTICE TO USERS WHO ARE MEMBERS OF THE CXL CONSORTIUM:

Members of the CXL Consortium (sometimes referred to as a "CXL Member") must be and remain in compliance with all of the following CXL Consortium documents, policies and/or procedures (collectively, the "CXL Governing Documents") in order for such CXL Member's use and/or implementation of this CXL Specification to receive and enjoy all of the rights, benefits, privileges and protections of CXL Consortium membership: (i) CXL Consortium's Intellectual Property Policy; (ii) CXL Consortium's Bylaws; (iii) any and all other CXL Consortium policies and procedures; and (iv) the CXL Member's Participation Agreement.

NOTICE TO NON-MEMBERS OF THE CXL CONSORTIUM, INC.:

If you are not a CXL Member and you have obtained a copy of this CXL Specification, you only have a right to review this document or make reference to or cite this document. Any references or citations to this document must acknowledge the Compute Express Link Consortium, Inc's sole and exclusive copyright ownership of this CXL Specification. The proper copyright citation or reference is as follows: "© 2023 COMPUTE EXPRESS LINK CONSORTIUM, INC. ALL RIGHTS RESERVED." When making any such citation or reference to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of the Compute Express Link Consortium, Inc.

Nothing contained in this CXL Specification shall be deemed as granting (either expressly or impliedly) to any party that is not a CXL Member: (ii) any kind of license to implement or use this CXL Specification or any portion or content described or contained therein, or any kind of license in or to any other intellectual property owned or controlled by the CXL Consortium, including without limitation any trademarks of the CXL Consortium; or (ii) any of the rights, benefits, privileges or protections given to a CXL Member under any CXL Governing Documents. For clarity, and without limiting the foregoing notice in any way, if you are not a CXL Member but still elect to implement this CXL Specification or any portion described herein, you are hereby given notice that your election to do so does not give you any of the rights, benefits, and/or protections of the CXL Members, including without limitation any of the rights, benefits, privileges or protections given to a CXL Member under the CXL Consortium's Intellectual Property Policy.

LEGAL DISCLAIMERS FOR, AND ADDITIONAL NOTICE TO, ALL PARTIES:

THIS DOCUMENT AND ALL SPECIFICATIONS AND/OR OTHER CONTENT PROVIDED HEREIN ARE PROVIDED ON AN "AS IS" BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, COMPUTE EXPRESS LINK CONSORTIUM, INC (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NON-INFRINGEMENT. In the event this CXL Specification makes any references (including without limitation any incorporation by reference) to another standard's setting organization's or any other party's ("Third Party") content or work, including without limitation any specifications or standards of any such Third Party ("Third Party Specification"), you are hereby notified that your use or implementation of any Third Party Specification: (i) is not governed by any of the CXL Governing Documents; (ii) may require your use of a Third Party's patents, copyrights or other intellectual property rights, which in turn may require you to independently obtain a license or other consent from that Third Party in order to have full rights to implement or use that Third Party Specification; and/or (iii) may be governed by the intellectual property policy or other policies or procedures of the Third Party which owns the Third Party Specification. Any trademarks or service marks of any Third Party which may be referenced in this CXL Specification is owned by the respective owner of such marks. The COMPUTE EXPRESS LINK®, CXL® and CXL LOGO trademarks (the "CXL

Contents

Revision History

Revision	Description	Date
1.0	First Release: H1-H12	December 13, 2023

H1 Clarifying ASPM is optional

In Section 10.4 CXL.io Link Power Management, *make the following update:*

CXL.io Link Power Management is as defined in PCIe Base Specification with the following notable differences:

- RCD links support ASPM-directed L1 entry but do not support PCI-PM-directed L1 entry. An eRCD is not required to initiate entry into L1 state when software transitions the device into D3Hot or D1 device state. When a component is not operating in RCD mode, the component shall support ~~ASPM L1 as well as~~ PCI-PM and optionally support ASPM L1. As such, a component not operating in RCD mode shall initiate CXL.io L1 entry when the device is placed in D3Hot or D1 device state
- ..

H2 Eliminate Device number 0 references

PCIe Base Specification requires that device capture the Device number instead of hard coding it to 0. Existing PCIe capabilities such as Flattening Portal Bridge and future PCIe capabilities may result in a component being assigned a non-zero Device number. This errata updates the text in CXL spec to be consistent with PCIe Base Specification. This errata does not apply to RCDs.

In Section 8.1.3 PCIe DVSEC for CXL Devices, make the following update:

..

A non-RCD is enumerated like a standard PCIe Endpoint and appears below a CXL Root Port or a CXL Switch. A non-RCD shall expose one PCIe device number and one or more function numbers at the parent Port's secondary bus number. These devices set PCI Express Capabilities Register.Device/Port Type=PCI Express* Endpoint and thus appear as standard PCIe Endpoints (EP). The PCIe Configuration Space of ~~Device 0,~~ Function 0 shall include the CXL PCIe DVSEC as shown in Figure 8-1.

In either case, the capability, status, and control fields in ~~Device 0,~~ Function 0 DVSEC control the CXL functionality of the entire device.

In Section 8.1.4 Non-CXL Function Map DVSEC, make the following update:

..

The PCIe Configuration Space of ~~Device 0,~~ Function 0 of a CXL device may include Non-CXL Function Map DVSEC as shown in Figure 8-2. ..

If this DVSEC capability is present, it must be included in ~~Device 0,~~ Function 0 of a CXL device and the Non-CXL Function Map bit corresponding to that Function shall be 0.

Absence of Non-CXL Function Map DVSEC indicates that PCIe DVSEC for CXL devices (Section 8.1.3) located on ~~Device 0,~~ Function 0 governs whether all Functions participate in CXL.cache and CXL.mem protocol.

8.1.4.1 Non-CXL Function Map Register 0 (Offset 0Ch)

Bit	Attributes	Description
31:0	HwInit	**Non CXL Function**: Each bit represents a non-virtual function number implemented by the device on the same bus as the physical function that carries PCIe DVSEC for CXL devices. When a bit is set, the corresponding Device/Function number or Function number (ARI device) is not capable of participating in CXL.cache or CXL.mem protocol. Bits corresponding to Non-existent Device/Function or Function numbers shall always return 0. If the device does not support ARI, bit x in this register maps to Device x, Function 0. If the device supports ARI, bit x in this register maps to Function x. ~~Bit 0 of this register shall always be cleared to 0 since PCIe DVSEC for CXL devices declares whether Device 0, Function 0 participates in CXL.cache and CXL.mem protocol.~~

In Section 8.1.8 PCIe DVSEC for Flex Bus Port, make the following update:

..

In RCHs and RCDs that implement RCRB, this DVSEC is accessed via RCRB.

The DVSEC associated with all other CXL devices shall be accessible via ~~Device 0,~~ Function 0 of the device.

In Section 8.1.9 Register Locator DVSEC, make the following update:

..

The PCIe Configuration Space of a CXL root port, CXL Downstream Switch Port, CXL Upstream Switch Port, and non-RCDs must implement this DVSEC capability. If a CXL device implements Register Locator DVSEC, it must appear in ~~Device 0,~~ Function 0 of the device. This requirement does not apply to CXL Switches.

In Section 8.1.11 Table Access DOE, make the following update:

..

Coherent Device Attributes Table (CDAT) allows a device or a switch to expose its performance attributes such as latency and bandwidth characteristics and other attributes of the device or the switch. A CXL Upstream Switch Port or ~~Device 0,~~ Function 0 of a CXL device may implement Table Access DOE capability, which can be used to read out CDAT, one entry at a time.

In Section 8.2.1.3 Flex Bus Port DVSEC, make the following update:

..

This DVSEC is also located in the Configuration Space of CXL root ports, Upstream Switch Ports, Downstream Switch Port, and CXL device's primary function (~~Device 0,~~ Function 0) if the device does not implement CXL RCRB. A CXL component that is neither an RCD nor an RCH shall report DVSEC Revision greater than or equal to 1. Revision 2 introduces 3 new registers.

H3 Misc ch 8 register definition updates

In 8.2.4.17.1 Uncorrectable Error Status Register (Offset 00h), make the following update:

Bit Location	Attribute	Description
..	..	..
6	RW1CS	Mem_BE_Parity: Internal Uncorrectable Byte Enable Parity error or other Byte Enable uncorrectable errors on CXL.mem. The Header Log register contains the M2S RwD Data Header if detected by either a host or a DSP. ~~The Header Log register contains the S2M DRS Data header if detected by either a device or a USP.~~ For CXL RAS Capability Version >=3, DWORD 0 of the Header Log register is reserved and the Data Header shall start at Byte Offset 4 of the Header Log register. For CXL RAS Capability Version <3, the position of the M2S RwD or S2M DRS Data Header in the Header Log register is not defined by this specification.
..	..	..

In 8.2.4.29.2 CXL Cache ID Decoder Control (Offset 04h), make the following update:

Bit Location	Attribute	Description
..	..	..

1	RW	**Assign Cache ID**: 1 indicates that this Downstream Port is either connected directly to a CXL.cache Device or the link is operating in 68B flit mode. In these cases, the Downstream Port~~and~~ assigns a Cache ID=Local Cache ID to it. The reset default is 0.
..	..	..

Update Table 8-86 as follows

Table 8-86. Media Test Results Long Log Entry Common Header

Byte Offset	Length in Bytes	Description
00h	1	**Number of Tests Executed**: Total number of tests executed by the device. This number can be smaller than the total number of tests that were requested.
01h	1	**Version**: Version of the Media Test Result Long Log. This field shall be set to 1. The version is incremented whenever the Media Test Results Long Log is extended to add more functionalities. Backward compatibility shall be maintained during this process. For all values of n, version n+1 may extend version n by replacing fields that are marked as reserved in version n or appending new fields but must not redefine the meaning of existing fields.
02h	1 ~~Eh~~	~~Reserved~~ Result: This field indicates the outcome of the execution of the requested list of tests. • 00h = All the tests completed successfully. • 01h = At least one test completed with a failure. • 02h = Test execution was interrupted by a Request Abort Background Operation command. All the tests that were complete, before the abort request was processed, ended successfully. • 03h = Test execution was interrupted by a Request Abort Background Operation command. At least one test completed with failure before the abort request ended. • All other encodings are reserved.
03h	0Dh	Reserved

Update Table 8-87 as follows

Table 8-87. Media Test Results Long Log Entry Structure

Byte offset	Length in Bytes	Description
00h	2	Test ID …
02h	8	Start Time …
0Ah	8	End Time …
12h	1	Result …
13h	1	Flags …
14h	4	Uncorrectable Error Count …
18h	4	Correctable Error Count …
1Ch	4	Reserved …
20h	8	Capacity Tested …
28h	4	Number of Error Signatures …
2Ch	4	Reserved
30h	~~20h~~ 50h	Error Signature 1
~~50h~~ 80h	~~20h~~ 50h	Error Signature 2
…	…	…
30h+((N-1)* ~~32~~50)h	~~20h~~ 50h	Error Signature N (N is the Number of error signatures)

*Update Table 8-19 Root Port n Security Policy Register (Offset 8*n-4) as follows*

Bit Location	Attributes	Description
1:0	RW	**Trust Level**: If the host supports only 1 CXL.cache device per VCS, this field defines the Trust Level for the CXL.cache Device below Root Port n (see Table 8-26 for definition of this field). If the host supports more than 1 CXL.cache device per VCS, this field defines the Trust Level that is applied to all ~~for~~ the CXL.cache devices below this root port ~~that is operating in HDM-D mode. The CXL Cache ID Decoder Control register (see Section 8.2.4.29.2) describes if such a device is present and if present, the associated Cache ID.~~ For an HDM-DB device, Trust Level=01 is equivalent to 00.

		Default value of this field is 10b.
~~2~~	~~RW~~	~~Block CXL.cache HDM-DB: This bit controls how the root port handles CXL.cache requests from the set of devices that are using HDM-DB flow. The CXL Cache ID Decoder Control register (see Section 8.2.4.29.2) identifies if a device below this root port is using in HDM-D flow and the associated Cache ID, if applicable.1~~ ~~• 0 = CXL.cache requests from any device using HDM-DB flow shall be permitted subject to other checks (see Section 9.15.2)~~ ~~• 1 = CXL.cache requests from any device using HDM-DB flow shall be unconditionally aborted by the root port~~ ~~Default value of this bit is 1.~~
31:~~3~~2	RsvdP	**Reserved**

~~1. This bit was introduced as part of Version=2.~~

Make the following changes to section 8.1.5.1

Bit	Attributes	Description
0	RO	**Port Power Management Initialization Complete**: When set, it indicates that the root port, the Upstream Switch Port or the Downstream Switch Port has successfully completed the Power Management Initialization Flow as described in Figure 3-4 and is ready to process various Power Management events. If this bit is not set within 100 ms of link-up, software may conclude that the Power Management initialization has failed and may issue Secondary Bus Reset to force link re-initialization and Power Management re-initialization. See Implementation Note.
13:1	RsvdP	**Reserved**
14	RW1CS	**Viral Status**: When set, indicates that the Upstream Switch Port or the Downstream Switch Port has entered Viral (see Section 12.4 for more details). This bit is not applicable to Root Ports, and reads shall return the value of 0.
15	RsvdP	**Reserved**

Implementation Note:

Certain conditions such as Link Down, Secondary Bus Reset or Downstream Port Containment reset the Downstream Component's bus number. If the Component generates CREDIT_RTN IP2PM message with Requestor Bus=0, the Downstream port may reject it if software has enabled ACS Source Validation. In this scenario, Power Management initialization may fail to complete and another Secondary Bus Reset alone will not facilitate recovery. Software may use the following sequence to recover from this failure

1. Save the ACS Source Validation bit and the Bus Master Enable bit in the Downstream Port
2. Clear Downstream Port's Bus Master Enable bit to 0.
3. Clear Downstream Port's ACS Source Validation bit to 0

4. Generate Secondary Bus Reset
5. Wait until the Port Power Management Initialization Complete bit is set in the Downstream Port
6. Restore the ACS Source Validation bit and the Bus Master Enable setting in the Downstream Port
7. Continue with device re-initialization

Update section 8.2.9.9.9.3 Add Dynamic Capacity Response (Opcode 4802h) as follows:

In response to an Add Capacity Event Record, or multiple Add Capacity Event records grouped via the More flag (see Table 8-50), the host is expected to respond with exactly one Add Dynamic Capacity Response acknowledgment, corresponding to the order of the Add Capacity Events received. If the order does not match, the device shall return Invalid Input. The Add Dynamic Capacity Response acknowledgement must be in sent in the same as order as the Add Capacity Event Records.

The device shall report Invalid Physical Address if:

- One or more extents in the updated extent list specify a DPA range that is outside the of range of the Extent List contained in the Add Capacity Event Record associated with this response. If the device receives an out of order Add Dynamic Capacity Response with non-zero Updated Extent List Size, it will result in this condition.
- One or more extents in the updated extent list specify a DPA range that has already been added with a previous call to Add Dynamic Capacity Response

Update section 8.2.9.9.9.4 Release Dynamic Capacity (Opcode 4803h) as follows:

.. This may be in response to a Release Capacity Event record or multiple Release Capacity Event records grouped via the More flag (see Table 8-50) , a Forced Release Capacity Event record or multiple Force Release Capacity Event records grouped via the More flag(see Table 8-50), or an unsolicited release of capacity that is not associated with an event. ..

Update Section 8.2.9.2.8 "Event Notification (Opcode 0106h) as follows

The recipient of this message may optionally acknowledge~~s~~ a notification by returning a response. ~~The component shall retransmit the notification every 1 ms using the same Message Tag value in the CCI Message (see Figure 7-19) until the recipient has returned a response with the Success return code, up to a maximum of 10 retries. No additional Event Notifications shall be sent until the component has received a response from the recipient.~~

Update Section 8.2.9.2.9 GFD Enhanced Event Notification (Opcode 0107h) as follows

The FM acknowledges a notification by returning a response. ~~The component shall resend the notification every 1 ms using the same Message Tag value in the transport header until the FM has~~

~~returned a response with the Success return code, up to a maximum of 10 retries. No additional Event Notifications shall be sent until the component has received a response from the FM.~~

H4 CXL.io Throttling Typo in Flit Type

This errata removes the CXL.io NOP Flit Type encoding referenced in the CXL.io Throttling feature description, as the value referenced was stale after the encoding definition was changed for CXL.io NOP.

Update section 6.4.3.1.2 as follows:

6.4.1.3.2 CXL.io Throttling

The Upstream Port must communicate to the Downstream Port during Phase 1 of alternate protocol negotiation if its CXL.io inbound path does not support receiving consecutive CXL.io flits (including CXL.io NOP flits with a DLLP payload) at a link speed of 64 GT/s. For the purpose of this feature, consecutive CXL.io flits are CXL.io Payload flits or CXL.io NOP flits with a DLLP payload ~~two flits with Flit Type encoding of 01b~~ that are not separated by either an intervening flit not associated with CXL.io~~with a different Flit Type encoding~~ or an intervening Ordered Set. Downstream Ports are required to support throttling transmission of CXL.io traffic to meet this requirement if the Upstream Port advertises this bandwidth limitation in the Modified TS1 Ordered Set (see **Table 6-9.**). One possible usage model for this is Type 3 memory devices that need 64 GT/s link bandwidth for CXL.mem traffic but do not have much CXL.io traffic; this feature enables such devices to simplify their hardware to provide potential buffer and power savings.

H5 Unexpected Flit Type Error in 256B Flit Mode

The current specification does not define how a receiver should handle a flit with Unexpected Flit Type in 256B Flit Mode. This errata specifies that an Unexpected Flit Type should be logged in the standard PCIe Flit Logging Extended Capability, in the Flit Error Log 1 Register, as an Unrecognized Flit.

Add section 6.2.3.3 as follows:

6.2.3.2 Framing Errors in 256B Flit Mode

An Unexpected Flit Type error is detected upon receiving a Flit with a Flit Type encoding associated with a Protocol that was not enabled during negotiation. For example, if a CXL.cachemem Flit Type is received while only CXL.io is enabled, this must be handled as an Unexpected Flit Type error. This is logged as an Unrecognized Flit in the PCIe Flit Logging Extended Capability, Flit Error Log 1 Register. Any interrupt signaling as a result of the logged error follows the PCIe specification definition.

Update Table 6-5 as follows to state CXL.cachemem flit type encoding is reserved if CXL.cachemem is not enabled:

Flit Header Field	Flit Header Bit Location	Description

Flit Type[1:0]	[7:6]	• 00b = Physical Layer IDLE flit or Physical Layer NOP flit or CXL.io NOP flit • 01b = CXL.io Payload flit • 10b = If CXL.cachemem is enabled, CXL.cachemem Payload flit or CXL.cachemem-generated Empty flit; reserved if CXL.cachemem is not enabled • 11b = ALMP Please refer to Table 6-6 for more details.
Prior Flit Type	[5]	• 0 = Prior flit was a NOP or IDLE flit (not allocated into Replay buffer) • 1 = Prior flit was a Payload flit or Empty flit (allocated into Replay buffer)
Type of DLLP Payload	[4]	• If (Flit Type = (CXL.io Payload or CXL.io NOP): Use as defined in PCIe Base Specification • If (Flit Type != (CXL.io Payload or CXL.io NOP)): Reserved
Replay Command[1:0]	[3:2]	Same as defined in PCIe Base Specification.
Flit Sequence Number[9:0]	{[1:0], [15:8]}	10-bit Sequence Number as defined in PCIe Base Specification.

H6 PID Interrupt Vector Typo and Chapter 7 Editorial Fixes

This errata fixes a typo in the PID Interrupt Vector definition in Table 7-160. This errata also makes some editorial fixes in Chapter 7.

Update section 7.7 as follows:

7.7 CXL Fabric Architecture

. . .

CXL fabric extensions allow for topologies of interconnected fabric switches using 12-bit PIDs (SPIDs/DPIDs) to uniquely identify up to 4096 Edge Ports. The following are the main areas of change to extend CXL as an interconnect fabric for server composability and scale-out systems:

• Expand the size of CXL fabric using Port Based Routing and 12-bit PIDs.

• Enable support for G-FAM devices (GFDs). A GFD is a highly scalable memory resource that is accessible by all hosts and all peer devices.

• Host and device peer communication may be enabled using UIO. ~~A future ECN is planned to complete the definition for this use case.~~

Update section 7.7.7.1 as follows:

. . .

Baseline Shared and Merged FC initialization and usage rules, as described in PCIe Base Specification, apply on ISLs as well, with some new rules/exceptions as noted below:

. . .

Update Table 7-160 in section 7.7.14.3 as follows:

Table 7-160. PID Interrupt Vector

Byte Offset	Length in Bytes	Description
000h	200h	**PID Interrupt Vector**: 4-Kb vector in which each bit corresponds to the associated PID (i.e., bit n represents PID n). A value of 1 in a bit position indicates that the GAE has received a~~n interrupt~~GAM VDM from the corresponding PID since the PID Interrupt Vector was last cleared.

H7 Responses for Requests Targeting NXM

The CXL specification is incomplete regarding CXL.mem requests targeting non-existent memory (NXM). It includes the MemData-NXM opcode for MemRd/MemRdData requests (that decode to non-existent memory) but does not mention how to handle the other request opcodes (e.g., MemInv). The new section, below, fits within Section 3.3 "CXL.mem" between current Section 3.3.10 and 3.3.11 and discusses the need for special handling while providing a table covering all opcodes. This errata also adds cross-references to this new section in existing tables: .

3.3.11 Responses for Requests Targeting NXM

Device responses to CXL.mem requests differ between HDM-H regions and HDM-D/HDM-DB regions, which creates an ambiguity when device receives a CXL.mem request it cannot map to a specific memory region. In this situation, devices shall respond according to Table 1-1. CXL.mem Responses for Requests to Non-existent Memory requesting device must accept and properly handle these responses regardless of its memory region decode results.

The ambiguity mentioned above is for reads and for some MemInv* cases. For reads, the response is DRS only for HDM-H or a DRS+NDR for HDM-D*. For MemInv*, HDM-H returns Cmp opcode and HDM-D/HDM-DB may expect only Cmp-E or Cmp-S as show in Table C-3 "HDM-D/HDM-DB Memory Requests".

The capability to support MemData-NXM is exposed in the "CXL HDM Decoder Capability Register" bit 20 (see Section 8.2.4.20.1).

Table 1-1. CXL.mem Responses for Requests to Non-existent Memory

CXL.mem Request	Device Response when NXM
MemRd, MemRdData, MemRdFill, MemRdTEE, MemRdDataTEE, MemRdFillTEE	MemData-NXM See Table 8-27 "CXL.mem Read Response – Error Cases" for additional details.
MemInv, MemInvNT, MemClnEvct, MemWr, MemWrPtl, MemWrTEE, MemWrPtlTEE	Cmp

Note: TEE requests have non-TEE response to allow requester to enforce appropriate security policy.

End of new section. The following are changes to cells in existing tables.

Table 3-53. S2M DRS Opcodes

Row "MemData-NXM", Column "Description" – Add cross reference to new section 3.13.

Table C-3. HDM-D/HDM-DB Memory Requests

Row "MemRd + MemData-NXM", Column "Description" – Add cross reference to new section 3.13.

Rows "MemInv", Column "Device Response, S2M NDR" – Add footnote to "Cmp-E" and "Cmp-S" cells with footnote indicating NXM case exception and cross reference to new section 3.13.

Row "MemRdData + MemData-NXM", Column "Description" – Add cross reference to new section 3.13.

Table C-6. HDM-H Memory Request

Footnote 2: Add cross reference to new section 3.13.

H8 Reserved Bit field forwarding

The CXL specification does not stated any requirement for Reserved bit forwarding in a switch. The new section below addresses the required handling for reserved bits. This fits within Section 7.3 "CXL.io, CXL.cachemem Decode and Forwarding" and under the sub-set for 7.3.2 CXL.cache and 7.3.3 CXL.mem.

7.3.2.3 CXL.Cache Reserved bit forwarding

A switch shall forward 256B Flit messages reserved bits between the ingress port and the egress port. Both HBR and PBR formats are defined for 256B flit messages where a switch can translate between those formats. When doing the translation between HBR and PBR formats defined for 256B flits the Reserved bits shall be preserved. When a switch with 256B flit capability sends to a port with 68B flit format the reserved bits shall be set to zero. Similarly, messages received as 68B flit formats shall never have reserved bits forwarded to a port with 256B flit messages.

Note: The reason for forwarding of reserved bits is to allow new features to be supported without requiring changes to existing switches. The reason for not forwarding in 68B flit format is that new features are expected to be added only to 256B flit formats so there is no need to support the complexity of translating reserved bits to/from 68B flits.

7.3.3.3 CXL.Mem Reserved bit forwarding

CXL.mem follows the same rules as CXL.cache as defined in Section 7.3.2.3.

H9 S2M Opcodes for 256B Flit only

The CXL.mem protocol has added new features that only apply to 256B flits. For M2S Req/RwD the opcode table notes the opcodes through use of a footnote. This was not done for S2M NDR/DRS messages and this errata adds the footnote to those opcode tables. Table 3-50 is in Section 3.3.9 "S2M No Data Response (NDR)" and Table 3-xx is in Section…

Note that the errata shows the foot note at the bottom of the page with opcode highlight, but when merged into the specification this will be attached to each table without the highlighting.

Table 3-50. S2M NDR Opcodes

Opcode	Description	Encoding
Cmp	Completions for Writebacks, Reads and Invalidates.	000b
Cmp-S	Indication from the DCOH to the Host for Shared state.	001b
Cmp-E	Indication from the DCOH to the Host for Exclusive ownership.	010b
Cmp-M	Indication from the DCOH to the Host for Modified state. This is optionally supported by host implementations and devices must support disabling of this response.	011b
BI-ConflictAck[1]	Completion of the Back-Invalidate conflict handshake.	100b
CmpTEE[1]	Completion for Writes (MemWr*) with TEE intent. Does not apply to any M2S Req.	101b
Reserved	Reserved	<Others>

Table 3-53. S2M DRS Opcodes

Opcode	Description	Encoding
MemData	Memory read data. Sent in response to Reads.	000b
MemData-NXM	Memory Read Data to Non-existent Memory region. This response is only used to indicate that the device or the switch was unable to positively decode the address of the MemRd as either HDM-H or HDM-D*. Must encode the payload with all 1s and set poison if poison is enabled. This special opcode is needed because the host will have expectation of a DRS only for HDM-H or a DRS+NDR for HDM-D*, and this opcode allows devices/switches to send a single response to the host, allowing a deallocation of host tracking structures in an otherwise ambiguous case.	001b
MemDataTEE[1]	Same as MemData but in response to MemRd* with TEE attribute.	010b
Reserved	Reserved	<Others>

[1] Only support in 256B flit mode.

H10 Chapter 7 Errata

In Section 7.2.1.3, make the following update:

In the case where the switch, FM, and host boot at the same time:

1. VCSs are statically defined.
2. DSP vPPBs within each VCS are unbound and presented to the host as Link Down.
3. Switch discovers downstream devices and presents them to the FM.
4. Host enumerates the VH and configures the DVSEC registers.

In Section 7.3.4, make the following update:

All PPBs are FM-owned. A PPB can be connected to a port that is disconnected or~~,~~ linked up~~ as an RCD, CXL SLD, or CXL MLD~~. SLD components can be bound to a host or unbound. Unbound SLD components can be accessed by the FM using CXL.io transactions via the FM API. LDs within an MLD component can be bound to a host or unbound. Unbound LDs are FM-owned and can be accessed through the switch using CXL.io transactions via the FM API.

In Section 7.5, make the following update:

Table 7-13. CXL Switch RAS

~~Host~~Triggering Action	Description	Switch Action for Non-pooled Devices	Switch Action for Pooled Devices
Switch boot	Optional power-on reset pin	Assert PERST# Deassert PERST#	Assert PERST# Deassert PERST#
Upstream PERST# asserted	VCS fundamental reset	Send Hot Reset	Write to MLD DVSEC to trigger LD Hot Reset of the associated LD **Note:** Only the FMLD provides the MLD DVSEC capability.
FM issues port reset command	Reset of an FM-owned DSP	Send Hot Reset	Send Hot Reset
PPB Secondary Bus Reset	Reset of an FM-owned DSP	Send Hot Reset	Write to MLD DVSEC to trigger LD Hot Reset of all LDs
USP receive~~d~~ Hot Reset	VCS fundamental reset	Send Hot Reset	Write to MLD DVSEC to trigger LD Hot Reset of the associated LD
USP vPPB Secondary Bus Reset	VCS US SBR	Send Hot Reset	Write to MLD DVSEC to trigger LD Hot Reset of the associated LD

DSP vPPB Secondary Bus Reset	VCS DS SBR	Send Hot Reset	Write to MLD DVSEC to trigger LD Hot Reset of the associated LD
Host writes FLR	Device FLR	No switch involvement	No switch involvement
Switch watchdog timeout	Switch fatal error	Equivalent to power-on reset	Equivalent to power-on reset

In Section 7.6.6.7, make the following update:

When a device is Hot-Added to an unbound port on a switch, the FM receives a notification and is responsible for binding as described in the steps below:

1. The switch notifies the FM by generating **Physical Switch Event Records** as the Presence Detect sideband signal is asserted or when a Link Up is detected if the PPB does not support Presence Detect~~and the port links up~~.

In Section 7.6.7.1.2, make the following update:

Table 7-20. Get Physical Port State Port Information Block Format

…		
5h	1	**Supported CXL Modes**: Formerly known as Connected CXL Version. Bitmask that defines which CXL modes are supported (1) or not supported (0) by this port: • Bit[0]: **RCD Mode** • Bit[1]: **CXL 68B Flit and VH Capable** • Bit[2]: **256B Flit** ~~and CXL~~ **Capable** • Bit[3]: **CXL Latency-Optimized 256B Flit Capable** • Bit[4]: **PBR Capable** • Bits[7:5]: Reserved for future CXL use Undefined when the value is 00h.
…		

In Section 7.6.7.2.1, make the following update:

Table 7-30. Get Virtual CXL Switch Info VCS Information Block Format

Byte Offset	Length in Bytes	Description
0h	1	**Virtual CXL Switch ID**
1h	1	**VCS State**: Current state of the VCS: • 00h = Disabled • 01h = Enabled • FFh = Invalid VCS ID; all subsequent field values are invalid • All other encodings are reserved
2h	1	**USP ID**: Physical port ID of the ~~local~~current VCS's Upstream Port, or the ~~local~~current VCS's fabric physical port ID of a Downstream ES VCS. Valid only when the VCS is enabled.
…		

In Section 7.6.7.3.1, make the following update:

When sent to an MLD, this provided command is tunneled by the FM-owned LD to the specified LD, as illustrated in the example in Figure 7-22 of a "Set LSA Request" being tunneled to LD 1 in an MLD.

In Section 7.6.7.6.1, make the following update:

Table 7-61. Get DCD Info Response Payload

Byte Offset	Length in Bytes	Description
00h	1	**Number of Hosts**: Total number of hosts that the device supports. This field shall have a minimum value of 1.
01h	1	**Number of Supported DC Regions**: The device shall report the total number of Dynamic Capacity Regions available per ~~host~~LD. DCDs shall report between 1 and 8 regions. All other encodings are reserved.
…		

In Section 7.6.7.6.5 Initiate Dynamic Capacity Add (Opcode 5604h), make the following update:

.. The processing of the actions initiated in response to this command may or may not result in a new entry or multiple entries grouped via the More flag (see Table 8-50) in the Dynamic Capacity Event Log. ..

In Section 7.7, make the following update:

- Host and device peer communication may be enabled using UIO. ~~A future ECN is planned to complete the definition for this use case.~~

In Section 7.7.11.5.1, make the following update:

The Link Partner Info payload includes the following fields about the far end of the link:

- 16B UUID
- 1B Physical Port
- 2B {4'b Device Type (0 = PBR switch, 1 = GFD, others reserved), 12b PBR ID (0xFFF is uninitialized)}
- 1B Standard FC VC list, default {8'bxxxx_0xx1}

In Section 7.7.13.6, make the following update:

Table 7-155. Get PID Binding Response Payload

Byte Offset	Length in Bytes	Description
0h	2	**PID**: PID of the remote binding target. 0xFFF if unbound.
2h	2	~~**Host ES Management PID**: Switch management PID of the Host ES. Valid only when the binding target is a Downstream ES VCS.~~**Reserved**
…		

In Section 7.7.13.7, make the following update:

Table 7-156. Configure PID Binding Request Payload

Byte Offset	Length in Bytes	Description
0h	1	• Bits[2:0]: **Operation**: — 000b = Bind — 001b = Unbind — All other encodings are reserved • Bits[7:3]: **Reserved**
1h	1	**Target VCS**: ID of the VCS to which the PID is being bound.
2h	1	**Target vPPB**: Index of the vPPB to which the PID is being bound. ~~Valid only~~Reserved when the binding target is a ~~Host~~Downstream ES VCS.
3h	1	**Reserved**
4h	2	• Bits [11:0]: **PID**: PID of the remote binding target. • Bits [15:12]: **Reserved**
6h	2	**Reserved**
8h	8	**Latency Entry Base Unit**: Latency Entry Base Unit for path between host and target device, as defined in ACPI HMAT System Locality Latency and Bandwidth Information Structure. Reserved~~Valid only~~ when the binding target is a ~~Downstream~~Host ES VCS.
10h	2	**Latency Entry**: Latency Entry for path between host and target device, as defined in ACPI HMAT System Locality Latency and Bandwidth Information Structure. Reserved~~Valid only~~ when the binding target is a Host~~Downstream~~ ES VCS.
12h	8	**BW Entry Base Unit**: Bandwidth Entry Base Unit for path between host and target device, as defined in ACPI HMAT System Locality Latency and Bandwidth Information Structure. Reserved~~Valid only~~ when the binding target is a Host~~Downstream~~ ES VCS.
1Ah	2	**BW Entry**: Bandwidth Entry for path between host and target device, as defined in ACPI HMAT System Locality Latency and Bandwidth Information Structure. Reserved~~Valid only~~ when the binding target is a Host~~Downstream~~ ES VCS.

In Section 7.7.14.1, make the following update:

Table 7-172. Identify GAE Response Payload

Byte Offset	Length in Bytes	Description
0h	2	• Bits[11:0]: **Total Number of Supported Enabled PIDs**: Maximum number of PIDs that can be enabled for concurrent use with the **Configure VCS PID Access** command • Bit[12]: **VendL0Prefix Enabled**: Indicates whether VendL0Prefix, if supported by the GAE, has been enabled • Bit[13]: **G-FAM/GIM Configuration Supported**: Indicates whether the switch supports (1) or does not support (0) re-configuration of the **GIM Support** bit with the **Set FAST Segment Entry** command • Bits[15:14]: **Reserved**
…		

H11 PBR Switch Port Processing of CXL Messages

Edit the following text as shown:

7.7.6.8 PBR Switch Port Processing of CXL Messages

…

In the CXL.io table (see Table 7-96), not all TLP types are explicitly covered; however, those not listed are usually handled by standard PCIe routing mechanisms (e.g., PCIe Messages are not explicitly covered, but ID-routed Messages are handled by PCIe ID routing, and address-routed Messages are handled by PCIe Memory Address routing). ~~Also, the UIO Direct P2P to HDM use case is not covered; see Section 7.7.7.~~

In the CXL.mem table (see Table 7-98) the Direct P2P CXL.mem to Accelerators use case is not covered; see Section 7.7.10.

Replace the existing PBR Switch Port Processing Tables with the following:

Table 7-96. PBR Switch Port Processing Table for CXL.io

Message Class & direction	Edge USP always below an RP	Host ES FPort with vDSP(s)	Downstream ES FPort with vUSP(s)	Edge DSP in either host ES or downstream ES			
				above HBR switch USP	above SLD	above MLD	above GFD
Cfg Req DS	PCIe ID routing to DSP or vDSP	vDSP converts to PBR fmt; FPort xmits to vUSP's FPort	vUSP matches DPID/SPID; vUSP converts to HBR fmt; PCIe ID routing to DSP	PCIe ID routing		PCIe ID routing LD-ID Prefix ←vPPB.LD-ID	N/A
Mem Req DS/US/P2P incl UIO DMA excl HDM UIO	PCIe Mem addr routing	DS: vDSP converts to PBR fmt; FPort xmits to vUSP's FPort US/P2P: vDSP matches DPID/SPID; vDSP converts to HBR fmt; PCIe Mem addr routing	DS: vUSP matches DPID/SPID; vUSP converts to HBR fmt; PCIe Mem addr routing US: vUSP converts to PBR fmt; FPort xmits to vDSP's FPort	PCIe Mem addr routing		PCIe Mem addr routing DS: LD-ID Prefix ←vPPB.LD-ID US/P2P: LD-ID Prefix identifies vPPB	N/A
Cpl US/P2P excl HDM UIO	PCIe ID routing	vDSP matches DPID/SPID; vDSP converts to HBR fmt; PCIe ID routing	vUSP converts to PBR fmt; FPort xmits to vDSP's FPort	PCIe ID routing		LD-ID Prefix identifies vPPB PCIe ID routing	N/A
Cpl DS excl HDM UIO	PCIe ID routing to DSP or vDSP	vDSP converts to PBR fmt; FPort xmits to vUSP's FPort	vUSP matches DPID/SPID; vUSP converts to HBR fmt; PCIe ID routing to DSP	PCIe ID routing		PCIe ID routing LD-ID Prefix ←vPPB.LD-ID	N/A
HDM UIO Req HDM Decoder case for Direct P2P & host requester	Direct P2P: N/A host req (DS): HDM Decoder routes to DSP or vDSP	DS: vDSP converts to PBR fmt; FPort xmits to vUSP's FPort US/P2P: vDSP matches DPID/SPID; vDSP converts to HBR fmt; USP's HDM Decoder routes P2P	DS: vUSP matches DPID/SPID; vUSP converts to HBR fmt; HDM Decoder routes to DSP US: vUSP converts to PBR fmt; FPort xmits to vDSP's FPort	US/P2P: if above MLD, LD-ID Prefix identifies vPPB; USP/vUSP HDM Decoder routes US or P2P within same switch DS: convert to HBR fmt; if above MLD, LD-ID Prefix ←vPPB.LD-ID			N/A
HDM UIO Cpl HDM Decoder case for Direct P2P & host requester	Direct P2P: N/A host req (US): PCIe ID routing	vDSP matches DPID/SPID; vDSP converts to HBR fmt; PCIe ID routing	vUSP converts to PBR fmt; FPort xmits to vDSP's FPort	US: if above MLD, LD-ID Prefix identifies vPPB; PCIe ID routing DS: PCIe ID routing; if above MLD, LD-ID Prefix ←vPPB.LD-ID			N/A
HDM UIO Req FAST/LDST case for Direct P2P & host requester	Direct P2P: N/A host req (DS): FAST/LDST converts to PBR and routes Eg2Eg	Route Eg2Eg	Route Eg2Eg	US/P2P: if above MLD, LD-ID Prefix identifies vPPB; FAST/LDST converts to PBR and routes Eg2Eg DS: convert to HBR fmt; if above MLD, LD-ID Prefix ←CAM$_{16}$(SPID)			US: N/A DS: keep in PBR
HDM UIO Cpl FAST/LDST case for Direct P2P & host requester	Direct P2P: N/A host req (US): convert to HBR	Route Eg2Eg	Route Eg2Eg	US: if above MLD, LD-ID Prefix identifies vPPB; convert to PBR using UIO ID-Based Rerouter; route Eg2Eg DS: convert to HBR			US: Keep in PBR; route Eg2Eg DS: N/A

Table 7-97. PBR Switch Port Processing Table for CXL.cache

Message Class & direction	Edge USP always below an RP	Host ES FPort with vDSP(s)	Downstream ES FPort with vUSP(s)	Edge DSP in either host ES or downstream ES			
				above HBR switch USP	above SLD	above MLD	above GFD
D2H Req US	Convert to HBR fmt CacheID←CAM_{16}(SPID)	Route Eg2Eg	Route Eg2Eg	Convert to PBR fmt DPID←vPPB.root.PID SPID←RT(CacheID)	Convert to PBR fmt DPID←vPPB.root.PID SPID←vPPB.self.PID		
H2D Rsp/DH DS	Convert to PBR fmt DPID←RT(CacheID)	Route Eg2Eg	Route Eg2Eg	Convert to HBR fmt 256B: CacheID←CAM_{16}(DPID) 68B: has no CacheID	Convert to HBR fmt 256B: CacheID←0 68B: has no CacheID		
H2D Req DS	Convert to PBR fmt DPID←RT(CacheID) SPID←vPPB.self.PID	Route Eg2Eg	Route Eg2Eg				
D2H Rsp/DH US	Convert to HBR fmt	Route Eg2Eg	Route Eg2Eg	Convert to PBR fmt DPID←vPPB.root.PID	Convert to PBR fmt DPID←vPPB.root.PID		

Table 7-98. PBR Switch Port Processing Table for CXL.mem

Message Class & direction	Edge USP always below an RP	Host ES FPort with vDSP(s)	Downstream ES FPort with vUSP(s)	Edge DSP in either host ES or downstream ES			
				above HBR switch USP	above SLD	above MLD	above GFD
M2S Req/RwD DS	FAST or LDST: Convert to PBR fmt DPID←xxST(HPA) SPID←vPPB.self.PID	Route Eg2Eg	Route Eg2Eg	Convert to HBR fmt LD-ID←0; is unused		LD-ID←CAM_{16}(SPID) Convert to HBR MLD fmt	LD-ID is N/A Keep in PBR fmt
	HDM Decoder: Convert to PBR fmt EvPPB←HDM-Dec(HPA) DPID←EvPPB.bndg.PID SPID←vPPB.self.PID Route to local DSP or vDSP FPort	vDSP's FPort xmits to vUSP's FPort	vUSP matches DPID/SPID vUSP keeps in PBR fmt EvPPB←HDM-Dec(HPA) DPID←EvPPB.self.PID Route to egress DSP				N/A
S2M NDR/DRS US	Convert to HBR fmt LD-ID←0; is unused	Route Eg2Eg	Route Eg2Eg	LD-ID is unused Convert to PBR fmt DPID←vPPB.root.PID		LD-ID identifies vPPB Convert to PBR fmt DPID←vPPB.root.PID	Keep in PBR fmt LD-ID is N/A
S2M BISnp US	Convert to HBR fmt BI-ID[11:0]←SPID	Route Eg2Eg	Route Eg2Eg	Convert to PBR fmt DPID←vPPB.root.PID BusNum←BI-ID[7:0] SPID←RAM_{256}(BusNum)	Convert to PBR fmt DPID←vPPB.root.PID SPID←vPPB.self.PID	BI-ID[3:0] contains LD-ID LD-ID identifies vPPB Convert to PBR fmt DPID←vPPB.root.PID SPID←vPPB.self.PID	Keep in PBR fmt
M2S BIRsp DS	Convert to PBR fmt DPID←BI-ID[11:0]	Route Eg2Eg	Route Eg2Eg	Convert to HBR fmt BusNum←RAM_{4k}(DPID) BI-ID[7:0]←BusNum	Convert to HBR fmt BI-ID is unused	Convert to HBR fmt LD-ID←CAM_{16}(SPID) BI-ID[3:0]←vPPB.LD-ID	Keep in PBR fmt

H12 PBR Support for UIO Direct P2P to HDM

Edit the following text as shown:

7.7.9 PBR Support for UIO Direct P2P to HDM

…

With LD-FAM devices, UIO completions can be routed edge-to-edge with an ID-Based Re-Router mechanism, which can be implemented in the Edge DSP above each LD-FAM device. The Re-Router matches against the ~~destination ID of~~ requester's PCI segment number (if applicable) and bus number in the UIO completion to determine the DPID for edge-to-edge routing. G-FAM devices automatically use edge- to-edge routing for UIO completions without this mechanism.

FAST decoders, LDST decoders, and ID-Based Re-Routers are each configured by host software using CCI command sets, as documented in Section ~~7.7.15~~ 7.7.14 for FAST decoders, and 7.7.13 for LDST decoders & ID-Based Re-Routers.