Errata for the Compute Express Link Specification Revision 3.2

March 17, 2025

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: "© 2025 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: I1-I13	December 20, 2024
2.0	Second Release: I14-I18	March 17, 2025

I1 L0p Responses After L0p Support Has Been Negotiated

A recently approved PCIe errata specifies that if the hardware autonomous width disable (HAWD) bit is set to 1 or L0p enable bit is cleared to 0 after L0p support was previously negotiated that the receiver of an L0p request is permitted in specific scenarios to not respond. This conflicts with the CXL specification as PCIe rules of abandonment are not applicable to CXL, and no response to an L0p request results in an uncorrectable internal error. This errata adds a clarifying statement to section 6.9 that was added in errata H13 to exclude no response from acceptable behavior.

6.9 L0p Support

CXL supports L0p Link width change as defined in the PCIe Base Specification with deltas specified in Section 5.1.2.5 and in this section.

The Hardware Autonomous Width Disable (HAWD) bit in the Link Control Register impacts both CXL.cachemem and CXL.io with the same effect as described in the PCIe specification~~.~~, except where the PCIe specification permits a receiver to not respond to an L0p request even after L0p support has been negotiated.

The L0p Enable bit in the Device Control 3 Register is defined in the PCIe Base Specification as determining Port behavior when sending or responding to Link Management DLLPs; for CXL, the L0p Enable bit determines Port behavior when sending or responding to L0p ALMPs for CXL.cachemem and for CXL.io. All other behavior described in the PCIe specification regarding this bit is applicable to CXL~~.~~, except where the PCIe specification permits a receiver to not respond to an L0p request even after L0p support has been negotiated.

I2 Inference of Electrical Idle

The PCIe specification states that while in L0, a device is permitted to infer Electrical Idle due to the absence of an UpdateFC DLLP (Optimized_Update_FC in Flit Mode) or SKP Ordered Set within a 128 us window; it permits a device to use the absence of either indicator or the absence of both indicators. In CXL, if CXL.io is in L1 or L2 while the Physical link is in L0, no UpdateFC DLLPs would be sent or received. This errata requires that CXL only use the SKP Ordered Set indicator for inference of Electrical Idle in L0; a statement is added in a new section 6.10.

6.10 Inference of Electrical Idle

Unlike PCIe where absence of UpdateFC DLLPs or SKP Ordered Sets may be used to infer Electrical Idle in L0, for CXL, only SKP Ordered Sets are permitted to be used for inferring Electrical Idle while in L0.

I3 Late Poison Clarifications

Add the following clarifications to 8.2.4.22.4 CXL IDE Error Status (Offset 0Ch):

Bit Location	Attributes	Description
3:0	RW1CS	**Rx Error Status**: Describes the error condition that transitioned the link to Insecure State if IDE stream is active. The component behavior upon this transition is defined in Section 11.3.8. • 0h = No Error • 1h = Integrity failure on received secure traffic • 2h = MAC or Truncated MAC received when the link is not in secure mode (when integrity is not enabled and the receiver detects MAC header) • 3h = MAC header received when not expected (No MAC epoch running, but the receiver detects a MAC header) • 4h = MAC header is not received when expected (MAC header not received within 6 flits after MAC epoch has terminated) • 5h = Truncated MAC flit is received when not expected (if the receiver gets truncated MAC flit corresponding to a completed MAC epoch) • 6h = After early MAC termination, the receiver detects a protocol flit (or Poison flit when IDE Protect Poison Message Enable is set) before the truncation delay • 7h = This error code encompasses the following conditions: — Protocol flit (or Poison flit when IDE Protect Poison Message Enable is set) received earlier than expected after key change (see Section 11.3.7 for the detailed timing requirements) — Rx IDE Stop.Enable=1 and a protocol flit (or Poison flit when IDE Protect Poison Message Enable is set) received earlier than expected after an IDE Termination Handshake (see Section 11.3.10 for the detailed timing requirements) • 8h = CXL.cachemem IDE Establishment Security error. This error code encompasses the following conditions: — IDE.Start is received prior to a successful CXL_KEY_PROG since the last Conventional Reset — IDE.Start is received prior to a successful CXL_KEY_PROG since the last IDE.Start — IDE.Start is received prior to a successful CXL_K_SET_GO since the last Conventional Reset — IDE.Start is received prior to a successful CXL_K_SET_GO since the last IDE.Start — CXL_IDE_KM message received over a different SPDM session (see Section 11.4.2) — IDE.Start is received in the middle of a MAC epoch (see Section 11.3.7) • All other encodings are reserved

Add the following new section 11.3.11.3 Error Reporting:

11.3.11.3 Error Reporting

Devices that are enabled for IDE Protection of the LLCTRL Poison Message and receive a poison control flit prior to the requisite truncation delay, shall report the error in the CXL IDE Error Status register by setting bit 06h or 07h, as appropriate, in the Rx Error Status field.

I4 Late Poison Offset and Definition Clarification

This errata cleans up the poison message offset definition for the late poison message; it also adds an additional example with a non-zero poison message offset. Additionally, this errata clarifies that the late poison message must only be used in scenarios where it is too late to utilize the poison bit in the header.

In Table 4-20 make the following edits:

In-band Error3	0011b	0001b	Poison	3:0	Poison Message Offset encodes the ~~is the encoding of which of the active or upcoming messages will have poison applied~~data message offset at which poison applies. There can be up to 8 ~~active~~ outstanding Data carrying messages ~~and up to 4 new data carrying messages~~ where the poison can be applied. • 0h = Poison the currently active data message • 1h = Poison the message 1 after the current data message • … • 7h = Poison the message 7 after the current data message See Section 4.3.6.3 for additional details.

In Section 4.3.6.3:

4.3.6.3 Late Poison

Late poison applies to cases where the header is sent already and the Tx intends to poison the corresponding data. For cases where header is not sent, Tx must poison the header if it intends to poison the corresponding data. Poison can be injected at a point after the header was sent by injecting an Error Control message with the Poison sub-type. The message includes a payload encoding that indicates the data message offset at which the poison applies. It is possible that any one of up to 8 active messages can be targeted. The encoding is an offset that is relative to the data that is yet to be sent, including the currently active data transmission. The poison applies to the entire message payload, just as it does when poison is included in the message header.

Add following non-zero poison offset example into section 4.3.6.3:

Example 3:

• Flit 1 - 1st 3 slots of data Message A in Slots 12 to 14.

• Flit 2 - In-band error poison message in Slot 0 with a poison message offset value of 1.

• Flit 3 - 4th slot of data Message A in Slot 1 and data Message B in Slots 2 to 5.

• The poison control message applies to Message B.

I5 Corrections to IDE Key Management

Update section 11.3.7 as follows:

..

After the keys are programmed into pending registers on both sides of the link, receipt of the CXL_K_SET_GO request shall cause each transmitter on each port to trigger the transmission of an IDE.Start Link Layer Control flit (see Table ~~4-3~~4-10 and 4-20).

..

Update Table 11-8 as follows:

Table 11-8. CXL_KEY_PROG Request

..	..	..
13h+KSIZE	12~~h~~	**Initial CXL.cachemem IDE IV**: Overwrites the Pending Initial *IV*. This field must be ignored if Use Default IV=1. Byte Offsets 16h+KSIZE:13h+KSIZE carry the *IV* DWORD, IV[95:64]. Byte Offsets 1A~~20~~h+KSIZE:17h+KSIZE carry the *IV* DWORD, IV[63:32]. Byte Offsets 1E~~24~~h+KSIZE:1B~~21~~h+KSIZE carry the *IV* DWORD, IV[31:0].
..	..	..

Update Table 11-18 as follows:

Table 11-18. CXL_GETKEY_ACK Response

..	..	..

13h+KSIZE	12	**Locally Generated CXL.cachemem IV**: This field must be ignored if the QUERY_RSP response message from the port indicates CXL.cachemem IV Generation Capable=0. Byte Offsets 16h+KSIZE:13h+KSIZE carry the *IV* DWORD, IV[95:64]. Byte Offsets 1A~~20~~h+KSIZE:17h+KSIZE carry the *IV* DWORD, IV[63:32]. Byte Offsets 1E~~24~~h+KSIZE:1B~~21~~h+KSIZE carry the *IV* DWORD, IV[31:0].
..	..	..

I6 Define CHMU Unit ID More Precisely

Update Section 8.2.8 "CHMU Register Interface" as follows:

..

If the counter related to a specific unit achieves the programmed hotness threshold during an epoch, the unit is considered hot, and is then reported to software. To report hot units to software, CHMU supports a circular structure called the CHMU Hotlist. The Hotlist is accessible through the CHMU's MMIO address space. There is a single Hotlist per CHMU instance for all the address ranges that can be configured on that CHMU. Each CHMU Hotlist entry is a 64-bit structure that contains a DPA unit address, called Unit ID, and its counter value. The Unit ID is the starting DPA address of the DPA unit, divided by the Unit Size. After the unit size is configured by the software, the remaining bits in the 64-bit entry can be allocated for the counters. Therefore, the hotness threshold can be configured only after setting the unit size. The size of the counters is retrieved through the CHMU Status register.

I7 CHMU Unit Size

*Update Section 8.2.8.3 "CHMU Configuration [i] (Offset 50h + CHMU Instance Length * i)", Table 8-37. "CHMU Configuration Register" as follows:*

..

Unit Size: Host-configured counting granularity. Configurable values are ~~expressed as~~ a power of 2 ~~and~~ ranging from 256B to 2 GB.

I8 Clarifications on PID assignments for Edge DSPs

A PBR Edge DSP requires at least one PID for all use cases and requires multiple PIDs for certain use cases. This erratum clarifies them.

Edit the following text as shown:

7.7.6.5 PID Use Models and Assignments

…

The Downstream ES FPorts may have one or more PIDs assigned, where each PID can be associated with a different set of FPorts. In an example scenario, there might be one PID for the left set of FPorts for multipathing and another PID for the right set. For a PID assigned to an FPort set for multipathing, DRTs in different USPs can specify different egress ports for static routing, distributing the static routing traffic for certain topologies without requiring additional DS_ES PIDs.

~~A DSP may be assigned multiple PIDs, one PID, or no PIDs. A DSP above a non-GFD usually has one PID, but may be assigned multiple PIDs for isolating traffic from multiple senders or for associating a unique PID for each caching or HDM-DB-capable device attached to one or more HBR switches below an Edge Port. DSPs above a multi-ported GFD may not require dedicated assigned PIDs, relying instead on one or more PIDs assigned to the GFD itself.~~

A DSP usually has one PID but may be assigned multiple PIDs when needed for CacheID or BI-ID translation. If one or more HBR switches below an Edge DSP attach multiple devices enabled for caching or multiple devices enabled for HDM-DB, then the Edge switch must assign multiple PIDs to the Edge DSP, so that each device assigned with a CacheID or BI-ID can be distinguished for proper PBR/HBR format translation at the Edge DSP.

I9 Clarifications for Configuring CacheID in PBR Fabric

A PBR Edge DSP may have multiple PIDs assigned for multiple use cases. This erratum elaborates on those use cases and clarifies the difference between non-GFD and GFD use cases.

Edit the following text as shown. Note that #3 is a new item that renumbered the items below it.

7.7.12.5 Configuring CacheID in PBR Fabric

From the host's perspective, configuration of CacheID for VHs spanning ~~a~~ multiple PBR ~~Fabric~~ switches is performed identically to such configuration in an exclusively HBR topology. PBR switches automatically internally configure and exchange ~~ID~~ CacheID/PID configuration information in the following manner:

1. The Host ES presents a Cache ID Route Table Capability in its Edge USP, and the Downstream ES (if present) presents ~~ID route table capabilities~~ a Cache ID Route Table Capability in its ~~vPPBs~~ vUSP (see Section 8.2.4.28 for details on the Cache ID Route Table).
2. The host will enumerate and assign all CacheIDs and program the ~~route table capability~~ Cache ID Route Table Capabilities, triggering the Commit bit to complete the configuration.
3. In a Host ES USP or Downstream ES vUSP, setting the Commit bit in its Cache ID Route Table Capability triggers the ES to configure its internal CacheID/PID mapping mechanism for any of its Edge DSPs that are mapped by its Cache ID Route Table entries.
4. ~~The setting of~~ In a Downstream ES, setting the Commit bit in a Cache ID Route Table Capability triggers the Downstream ES to generate one or more RTUpdate VDMs, as defined in Section 3.1.11.7, ~~targeted at~~ targeting the Host PID. The Host ES will intercept this VDM based on its PBR opcode.
5. Upon receipt of the VDM, the Host ES programs the necessary CacheID to PID translation logic in the Host edge port.
6. The Host ES acknowledges successful programming of the ~~ID~~ CacheID/PID translation logic with an RTUpdateAck VDM, as defined in Section 3.1.11.8, sent to the Downstream ES for each RTUpdate VDM that was received and successfully processed.
7. Upon receipt of the VDM, the Downstream ES sets the corresponding 'RT Committed' bit in the vUSP.

An ~~downstream~~ HBR switch topology below an Edge DSP (in either a Host ES or Downstream ES) requires PIDs for each unique potential target so that PID/CacheID translation can occur at that Edge DSP ~~IDs can be translated between CacheID and PID at the fabric edges~~. For CacheID, the ~~ID~~ translation is valid if the Valid bit is set in a Cache ID Target entry in the Cache ID Route Table Capability Structure. The corresponding PID used is the PID of the DSP to which the Route Table entry has been configured to map. Multiple PIDs must be assigned to a DSP if multiple CacheIDs map to targets below that DSP.

I10 DSAR bit value rule fixes in Section 7.7.7.1

The DSAR bit value rules in Section 7.7.7.1 have several problems that can result in PBR fabric deadlocks. One involves GIM responses. A second one involves P2P traffic with CXL.io TLPs in general. A third one involves TLPs that originate within the PBR fabric.

Edit the following text as shown:

7.7.7.1 .io Deadlock Avoidance on ISLs/PBR Fabric

ISLs and PBR switches carry CXL.io Upstream traffic and CXL.io Downstream traffic from different hosts in the same physical direction/queues. To avoid deadlocks, these two traffic types need to be kept independent on ISLs and internally through PBR switches. To assist in maintaining the required independence, each TLP inside the PBR fabric is tagged with a DSAR (Downstream Acceptance Rules) bit. Here are the rules for setting the value of the DSAR bit within the PTH:

- When an Edge DSP converts a received TLP from HBR to PBR format, the Edge DSP shall clear the DSAR bit
- When an Edge USP converts a received TLP from HBR to PBR format, the Edge USP shall clear the DSAR bit if the TLP is a UIO completion with VendPrefixL0, and set the DSAR bit in all other cases
- When a ~~Host ES vDSP~~ switch VCS forwards a TLP within its virtual hierarchy P2P (from coming upstream to going downstream), ~~it~~ the VCS shall set the forwarded TLP's DSAR bit
- When a function in a PBR switch originates a CXL.io request or completion TLP that travels upstream, it shall clear the DSAR bit. If the TLP travels downstream, the function shall set the DSAR bit
- When a GFD sends a TLP (which is always in PBR format), the GFD shall clear the DSAR bit
- When an Edge DSP above a GFD forwards a TLP to the GFD, the Edge DSP shall set the DSAR bit

For the remainder of this section, traffic with DSAR=0 is referred to as USAR (Upstream Acceptance Rules) traffic, and DSAR=1 traffic is referred to as DSAR (Downstream Acceptance Rules) traffic. On an ISL, this bit is carried in the PTH. Traffic within each VC is required to follow the ordering rules specified in Table 7-107 and Table 7-108.

I11 Sticky Register Test

Update section 14.15.1 with the following to match the register settings in 8.2.4.16.2 and 8.2.4.16.3:

14.15.1 Sticky Register Test

…

Uncorrectable Error Mask Register (Offset 04h)

Bits	Variable	Settings

~~12~~11:0	Error Mask Registers	Set to ~~1~~FFFh
~~15~~14:~~15~~14	Internal_Error_Mask	Set to 1
~~16~~15:~~16~~15	CXL_IDE_Tx_Mask	Set to 1
16:16	CXL_IDE_Rx_Mask	Set to 1

…

Uncorrectable Error Severity Register (Offset 08h)

Bits	Variable	Settings
~~12~~11:0	Error Severity Registers	Set to FFFh
~~15~~14:~~15~~14	Internal_Error_Severity	Set to 1
~~16~~15:~~16~~15	CXL_IDE_Tx_Severity	Set to 1
16:16	CXL_IDE_Rx_Severity	Set to 1

…

Note, for 2.0 spec, these registers are defined in 8.2.5.9.2 and 8.2.5.9.3

I12 14.6.9 Link Speed Degradation Below 8GT/s

Update section 14.6.9 with the following

Test Steps:

1. Train the CXL link up to the highest speed possible (at least 8GT/s)

2. Degrade the link down to a speed below CXL mode operation by requesting speed change to data rate below 8GT/s using Recovery State machine

~~3. Link goes to detect state~~

Pass Criteria:

• Link degrades to slower speed and operate in PCIe mode, or

• Link enter Detect if link cannot recover. (One example case of link cannot recover is test equipment's DSP staying at EI long enough to force DUT's USP to enter Detect per LTSSM requirement.)

Fail Criteria:

• Link stays in CXL mode

• Link does not change speed

I13 CHMU Epoch Restart

Update Section 8.2.8 "CHMU Register Interface" as follows:

..

When Epoch-based reporting mode is enabled, hot units are reported in the CHMU Hotlist at the end of their counter's current epoch. Even if the counter reaches the hotness threshold before the epoch ends, the counter continues counting until the epoch ends. Consequently, the counter values in the CHMU Hotlist entries will contain the number of hot unit accesses within the epoch. The counters within the counting structure will be freed at the end of their respective epochs.

When Always-on reporting mode is enabled, after the counters achieve the hotness threshold, their corresponding Unit ID is immediately reported in the CHMU Hotlist and becomes visible to the software through the CHMU Hotlist-related registers. The hot unit-related counters are freed when the counters are reported in the Hotlist. The counters within the counting structure will be freed at the end of their respective epochs.

Epochs can be global or per-counter as specified by the Epoch Type field in the CHMU Capability Register. When a global epoch ends, a new epoch starts immediately after freeing all counters used in the epoch. When a per-counter epoch ends, a new epoch for that counter will start when it is allocated again.

CHMU supports a down-sampling factor for the incoming M2S requests that allows sampling at a configurable rate, or at a rate that is controlled by the device. The down-sampling factor can be either selectable by software (if supported by the device) or can be selected by the device.

…

Update Section 8.2.8.2 "CHMU Capability Register (Offset 10h + CHMU Instance Length * i1)" Table 8-36:

Epoch Type:

• 00b = Global. All counters start and end their epoch simultaneously.

• 01b = Per counter. Each counter starts and ends their epoch independently. The ~~counter~~ counter's epoch starts when it is allocated.

• All other encodings are Reserved.

I14 NOP Alignment and Max TLP Clarification

This errata clarifies how the NOP TLP alignment rules and the maximum non-NOP TLP packing rules from the PCIe specification are applied to CXL standard and latency-optimized 256B flits.

Make the following updates in Section 4.1 of the Link Layer chapter:

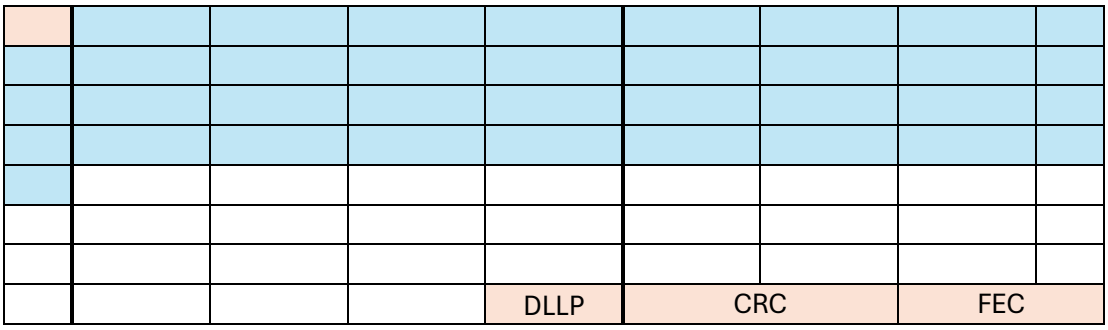
For 256B Flit mode, the NOP-TLP alignment rules from the PCIe Base Specification for PCIe Flit mode are shifted as a result of the two bytes of Flit ~~Type~~Header at the beginning of the flit. The NOP TLP alignment is additionally shifted by the six bytes of CRC at the end of the first flit half in latency-optimized 256B flits. In other words, NOP TLP alignment must be calculated relative to the TLP payload, which is the 236 bytes of Flit Data in standard 256B flits and the 232 bytes of Flit Data in latency-optimized 256B flits. See Figures 4-<TBD1> and 4-<TBD2> where the thick black lines are the NOP-TLP alignment boundary.

The maximum non-NOP TLP rules from the PCIe Base Specification are applied to CXL flits as described below: (see Figures 4-<TBD1> and 4-<TBD2> where the blue and white groups represent the two groups described in the below bullet items)

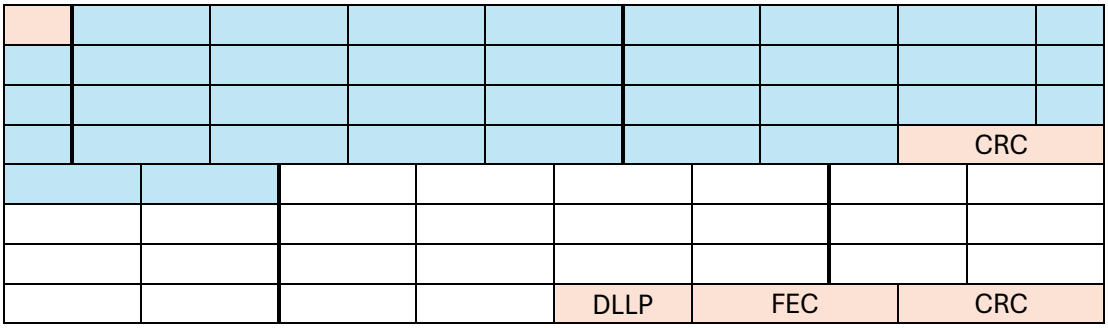
- For standard 256B flits, the rules are applied independently to the following two groups of bytes: 1) the first 128 TLP bytes and 2) the remaining 108 TLP bytes.
- For latency-optimized 256B flits, the rules are applied independently to the following two groups of bytes: 1) the first 128 TLP bytes and 2) the remaining 104 TLP bytes.
- For each of the above groups, the PCIe rule of no more than 8 non-NOP TLPs, including partial TLPs applies.

Note, depending on implementation choices on the receive side, the logic may have to handle processing more than 8 non-NOP TLPs in a clock cycle.

Add new Figure 4-<TBD1>: Standard 256B Flit NOP Alignment and Max TLP



Add new Figure 4-<TBD2>: Latency-Optimized 256B Flit NOP Alignment and Max TLP



Add cross reference to end of Section 3.1 of the Transaction Layer chapter:

Refer to section 4.1 for details on NOP TLP alignment rules and maximum non-NOP TLP packing rules.

I15 Restrictions with Host-to-Host UIO Usages

This section has major problems with its characterization of the deadlock issues and its suggested avoidance approaches. This erratum corrects the characterization of the deadlock issues and offers alternative workarounds.

Edit the following text as shown:

7.7.3.4 Restrictions with Host-to-Host UIO Usages

GIM is a standardized mechanism supporting Host-to-Host (H2H) UIO DMA. However, implementations may leverage PBR fabric mechanisms architected for GIM to support other (including proprietary) H2H protocols that use UIO. This section covers restrictions that apply to both GIM and H2H UIO communications in general.

~~Host-to-Host~~ H2H UIO usages can result in deadlock ~~when mixed with UIO traffic going to the host that can route back in the host.~~ if a host introduces shared resources between the following two roles. Refer to the Implementation Note below to see an example deadlock.

- H2H responder role: The host is an H2H UIO responder.
- UIO forwarding role: The host performs any operation where a given UIO TLP is ingressed at an RP and later egressed at the same RP or another RP. This includes forwarding H2H or non-H2H TLPs, as well as reflecting TLPs sent to the RC for validation and/or remapping, e.g., by ACS P2P Redirect mechanisms.

There are different approaches to avoid such deadlocks. For example, with a host that plays a H2H responder role, ensure that the host never plays a UIO forwarding role. More specifically, ensure that no devices send UIO TLPs that require forwarding or reflecting by their local host. Note that:

- UIO Direct P2P to HDM traffic within a PBR fabric requires no host forwarding.
- H2H UIO traffic originated by host CPU load/store accesses requires no host forwarding.
- UIO traffic by devices to resources within their local host (e.g., for local DMA) requires no host forwarding.
- H2H UIO traffic by devices to remote hosts (e.g., using GIM) does require their local host to perform H2H UIO forwarding. This is not supported if the device's local host is concurrently playing the H2H responder role.
- UIO P2P traffic between devices that is forwarded by a host's RPs may be challenging to identify and avoid, e.g., any UIO P2P traffic redirected by ACS P2P Redirect mechanisms. Regardless, this is not supported if the host is concurrently playing the H2H responder role.
- ~~Systems that support Host-to-Host UIO must use a separate VC for Host-to-Host UIO traffic vs. remainder of UIO, on host edge links.~~

~~(OR)~~

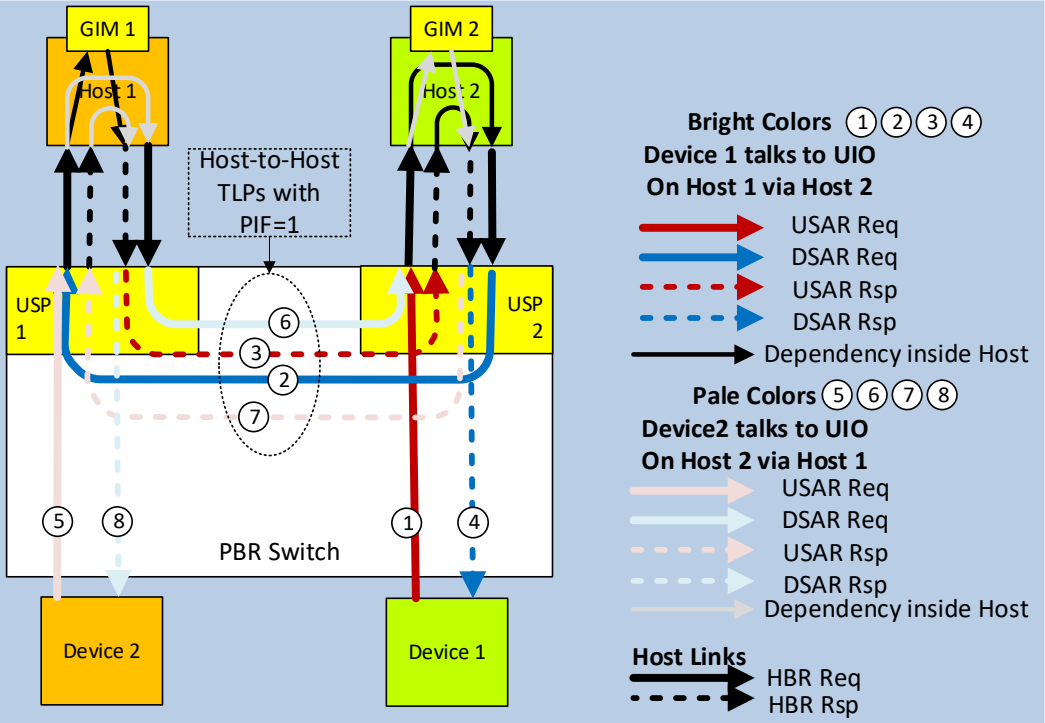
- ~~Minimally avoid usages that can cause loopback traffic, either in the host or in switches. Generically, this restriction could mean that UIO accesses do not target MMIO space.~~

~~A detailed analysis of restrictions that are needed to make a specific system configuration to work with Host-to-Host UIO enabled is beyond the scope of this specification.~~

A future ECN may ~~be considered that allows for more~~ formalize deadlock avoidance options beyond ~~the two~~ those listed above.

IMPLEMENTATION NOTE

Here is an example deadlock with GIM, based on circular request dependencies that arise when devices under multiple GIM hosts are accessing remote GIM concurrently. In this example, circles 1-4 represent UIO TLPs in the path when device 1 accesses GIM 1 on host 1. Circles 5-8 represent UIO TLPs in the path when device 2 accesses GIM 2 on host 2.



In this example, UIO Request (Req) dependencies form a closed loop of head-of-line (HOL) blocking instances, specifically:

- At USP 1 egress, USAR Req 5 blocks DSAR Req 2 due to HOL blocking
- At host 2 egress, DSAR Req 2 blocks USAR Req 1 due to HOL blocking
- At USP 2 egress, USAR Req 1 blocks DSAR Req 6 due to HOL blocking
- At host 1 egress, DSAR Req 6 blocks USAR Req 5 due to HOL blocking
- This completes the circle, resulting in deadlock

In each HOL blocking case, the fundamental cause is comingling USAR & DSAR Requests in the same flow control class traversing an HBR link.

A similar example exists with circular UIO Response (Rsp) dependencies.

I16 TSP Updates – Remove CXL Timeout and Isolation Capability Structure

Collection of TSP bugs for the CXL 4.0 specification.

11.5.4.8.1 Locking the Target

- Altering link or Transaction Layer behavior at runtime

— PCIe DVSEC for Flex Bus Port registers.
~~— CXL Timeout and Isolation Control register.~~
-CXL IDE Control register …

11.5.4.8.2 Considerations for Securing the Host

— Altering link or Transaction Layer behavior at runtime

- PCIe DVSEC for Flex Bus Port registers
- CXL Timeout and Isolation Control register.
- CXL IDE Control register…

11.5.5.6.6 Get Target Configuration Report Response

Table 11-41. TSP Report

		Valid TSP Report Fields: More than one bit may be set. • Bit[0]: **CXL IDE Capability Structure Valid**: When set, the TSP Report contains a valid CXL IDE Capability Structure • Bits[7:1]: **Reserved**
01h	3	**Reserved**
04h	3Ch	**PCIe DVSEC for CXL Devices**: See Section 8.1. Reports the first 3Ch bytes of the structure.
40h	20h	**PCIe DVSEC for Flex Bus Port**: See Section 8.2. Reports the first 20h bytes of the structure.
60h	38h	**CXL Link Capability Structure**: See Section 8.2.4.19. Reports the first 38h bytes of the structure.
98h	10h	~~**CXL Timeout and Isolation Capability Structure**: See Section 8.2.4.24.~~ **Reserved**
A8h	10h + k*20h	**CXL HDM Decoder Capability Structure**: The number of HDM decoders (k) is specified in the Decoder Count field in the CXL HDM Decoder Capability register. See Section 8.2.4.20. Reports the first 20h bytes of the structure.
B8h + k*20h	24h	**CXL IDE Capability Structure**: Structure for optionally reporting the CXL IDE Transport Security configuration. Valid if CXL IDE Capability Structure Valid is set. Reports the first 24h bytes of the structure.

I17 TSP Updates – Correction to TSP Compliance Test

14.11.7.12 Target-based range-based memory encryption

This test verifies basic setting and clearing of encryption keys for optional target-based range-based memory encryption.

Prerequisites:

- Device must support CXL.cachemem TSP security
- Device must support Compliance Mode DOE and SPDM over DOE

• Host software has established a secure SPDM link to the device

• Test 14.11.7.3 passed AND the target reports support for target-based Rangebased Encryption

Topologies:

• SHDA

Test Steps:

1. Host software issues Set Target Configuration to enable target-based range-based memory encryption

a. Memory Encryption Features Enable flags, Encryption set, Range-based Encryption set

b. Memory Encryption Algorithm Select has a single algorithm selected that the target will utilize for data-at-rest security. Shall be one of the algorithms supported by the target as reported by Get Target Capabilities.

2. Host software receives Set Target Configuration Response

3. Host software issues Lock Target Configuration to make the configuration immutable

4. Host software receives Lock Target Configuration Response

5. Host software issues Set Target Range Specific Key to associate a key with the first memory range

a. Range ID = 0

b. Range Start and Range End describe the test address range on the host

c. Validity Flags, Bit[0] set

d. Data Encryption Key to utilize for the memory Range ID

6. Host software receives Set Target Range Specific Key Response

7. Host software generates a full cache-line memory write request for the test address range using a known data pattern A

8. Host software generates a memory read request to the same address

9. Host software verifies the data matches the known data pattern A

10. Host software issues Clear Target Range Specific Key to remove the association of a key with the memory range

a. Range ID = 0

11. Host software receives Clear Target Range Specific Key Response

12. Host software generates a full cache-line memory write request for the test address

range using a known data pattern B

13. Host software generates a memory read request to the same address

14. Host software verifies the data does NOT match the known data pattern A or B

Pass Criteria:

- Data does not match expected pattern after any reads

Fail Conditions:

- Set Target Configuration results in a TSP Error Response
- Lock Target Configuration results in a TSP Error Response
- Set Target Range Specific Key results in a TSP Error Response
- Clear Target Range Key results in a TSP Error Response
- Pass criteria is not met

I18 Replace the word Hotness with Hot-range

This errata uses a more neutral word choice to replace an existing word that was flagged as having dual meanings.

Throughout the specification, replace all instances of the word “Hotness” with the term “Hot-range”.