Compute Express Link™ (CXL™)

Errata for the Compute Express Link Specification Revision 2.0

May 2021

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

This CXL **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:

If you are a Member of the CXL Consortium (sometimes referred to as a "**CXL Member**"), and even if you have received this publicly-available version of this CXL Specification after agreeing to CXL Consortium's Evaluation Copy Agreement (a copy of which is available https://www.computeexpresslink.org/download-the-specification, each such CXL Member must also be 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:

If you are **not** a CXL Member and have received this publicly-available version of this CXL Specification, your use of this document is subject to your compliance with, and is limited by, all of the terms and conditions of the CXL Consortium's Evaluation Copy Agreement (a copy of which is available at https://www.computeexpresslink.org/download-the-specification**).**

In addition to the restrictions set forth in the CXL Consortium's Evaluation Copy Agreement, 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: **"© 2019-2021 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.

Except for the limited rights explicitly given to a non-CXL Member pursuant to the explicit provisions of the CXL Consortium's Evaluation Copy Agreement which governs the publicly-available version of this CXL Specification, 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 benefits and/or rights as a CXL Member under any CXL Governing Documents.

LEGAL DISCLAIMERS FOR ALL PARTIES:

THIS DOCUMENT AND ALL SPECIFICATIONS AND/OR OTHER CONTENT PROVIDED HEREIN IS 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.

NOTICE TO ALL PARTIES REGARDING THE PCI-SIG UNIQUE VALUE PROVIDED IN THIS CXL SPECIFICATION:

NOTICE TO USERS: THE UNIQUE VALUE THAT IS PROVIDED IN THIS CXL SPECIFICATION IS FOR USE IN VENDOR DEFINED MESSAGE FIELDS, DESIGNATED VENDOR SPECIFIC EXTENDED CAPABILITIES, AND ALTERNATE PROTOCOL NEGOTIATION ONLY AND MAY NOT BE USED IN ANY OTHER MANNER, AND A USER OF THE UNIQUE VALUE MAY NOT USE THE UNIQUE VALUE IN A MANNER THAT (A) ALTERS, MODIFIES, HARMS OR DAMAGES THE TECHNICAL FUNCTIONING, SAFETY OR SECURITY OF THE PCI-SIG ECOSYSTEM OR ANY PORTION THEREOF, OR (B) COULD OR WOULD REASONABLY BE DETERMINED TO ALTER, MODIFY, HARM OR DAMAGE THE TECHNICAL FUNCTIONING, SAFETY OR SECURITY OF THE PCI-SIG ECOSYSTEM OR ANY PORTION THEREOF (FOR PURPOSES OF THIS NOTICE, "**PCI-SIG ECOSYSTEM**" MEANS THE PCI-SIG SPECIFICATIONS, MEMBERS OF PCI-SIG AND THEIR ASSOCIATED PRODUCTS AND SERVICES THAT INCORPORATE ALL OR A PORTION OF A PCI-SIG SPECIFICATION AND EXTENDS TO THOSE PRODUCTS AND SERVICES INTERFACING WITH PCI-SIG MEMBER PRODUCTS AND SERVICES).

Contents

Revision History

Revision	Description	Date
1.0	First Release: Errata F1-F33	May 5, 2021

F1 ARB/MUX State Transition Table

In section 5.1, Table 60, make the following changes to delete the Active->Reset transition and add the Retrain to Reset transition:

Table 60. ARB/MUX State Transition Table

Current vLSM State	Next State	Upstream Port Trigger Condition	Downstream Port Trigger Condition
Active	L1.x	Upon receiving a Request to enter L1.x from Link Layer, the ARB/MUX must initiate a Request ALMP{L1.x} and receive a Status ALMP{L1.x} from the remote vLSM	Upon receiving a Request to enter L1.x from Link Layer and receiving a Request ALMP{L1.x} from the Remote vLSM, the ARB/MUX must send Status ALMP{L1.x} to the remote vLSM
	L2	Upon receiving a Request to enter L2 from Link Layer the ARB/MUX must initiate a Request ALMP{L2} and receive a Status ALMP{L2} from the remote vLSM	Upon receiving a Request to enter L2 from Link Layer and receiving a Request ALMP{L2} from the Remote vLSM the ARB/MUX must send Status ALMP{L2} to the remote vLSM
	~~Reset~~	~~Physical Layer transitions from Recovery to L0 and State Status ALMP synchronization for Recovery exit resolves to Reset. (see Section 5.1.2.3)~~	~~N/A~~
L1	Retrain	Upon receiving an ALMP Active request from remote ARB/MUX	Upon receiving an ALMP Active request from remote ARB/MUX
Active	Retrain	Any of the following conditions are met: 1) Physical Layer LTSSM enters Recovery.	Physical Layer LTSSM enters Recovery.

		2) Physical Layer transitions from Recovery to L0 and State Status ALMP synchronization for Recovery exit resolves to Retrain. (see Section 5.1.2.3)	
Retrain	Active	Link Layer is requesting Active and any of the following conditions are met: 1) Physical Layer transitions from Recovery to L0 and State Status ALMP synchronization for Recovery exit resolves to Active. 2) Physical Layer transitions from Recovery to L0 and State Status ALMP synchronization for Recovery exit does not resolve to Active. Entry to Active ALMP exchange protocol is complete. (See Section 5.1.2.2.) 3) Physical Layer has been in L0. Entry to Active ALMP exchange protocol is complete. (See Section 5.1.2.2.)	Link Layer is requesting Active and any of the following conditions are met: 1) Physical Layer transitions from Recovery to L0 and State Status ALMP synchronization for Recovery exit resolves to Active. 2) Physical Layer transitions from Recovery to L0 and State Status ALMP synchronization for Recovery exit does not resolve to Active. Entry to Active ALMP exchange protocol is complete. (See Section 5.1.2.2.) 3) Physical Layer has been in L0. Entry to Active ALMP exchange protocol is complete. (See Section 5.1.2.2)
Retrain	Reset	Physical Layer transitions to L0 and State Status ALMP synchronization for Recovery exit resolves to Reset. (see Section 5.1.2.3)	N/A

F2 PCIe Capable/Enable Description in Table 68

In section 6.3.1.1, Table 68, make the following changes:

Table 68. Additional Information on Symbols 12-14 of Modified TS1/TS2 Ordered Sets

Bit Field in Symbols 12-14	Description
PCIe capable/enable	The Downstream Port and Upstream Port advertise their capability in Phase 1~~ as set in the Flex Bus Port Control register in Section 8.2.1.3.2~~. The Downstream Port communicates the results of the negotiation in Phase 2.[1]

F3 Making Hot-plug support optional for Downstream Ports

In section 9.9, make the following changes:

~~CXL 2.0 Root Ports, CXL 2.0 devices and CXL switches shall support Hot-Add and managed Hot-Remove.~~ CXL 2.0 Root Ports and CXL Downstream Switch Ports may support Hot-Add and managed Hot-Remove. All CXL 2.0 Ports shall be designed to avoid electrical damage upon surprise Hot-Remove. CXL 2.0 Devices and Switches shall be capable of being Hot-plugged, subject to the Form Factor limitations.

F4 Chapter 8 Miscellaneous errata

In section 8.1.2, make the following addition to the footnote below Table 125. CXL DOE Type Assignment:

Support for the Compliance DOE Type is highly recommended for CXL 2.0 devices. If Compliance DOE Type is not implemented by a device, it shall implement PCIe DVSEC for Test Capability (Section 14.16.1). PCIe DVSEC for Test Capability is not required for CXL 2.0 devices running in CXL 1.1 mode if they implement Compliance DOE Type.

In section 8.1.3.3, make the following change:

Bit	Attributes	Description
13:0	Revds.	Reserved.
14	RW1CS	Viral_Status: When set, indicates that the CXL device ~~has~~had encountered a Viral condition~~entered Viral~~. ~~Viral.~~ This bit does not

		indicate that the device is currently in Viral condition. See Section 12.4, "CXL Viral Handling" for more details.
15	RsvdZ	Reserved.

In section 8.1.4.1, make the following change:

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 function that carries PCIe DVSEC for CXL Device. 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 set to 0~~1~~ since PCIe DVSEC for CXL Device declares whether Device 0, Function 0 participates in CXL.Cache and CXL.Mem protocol.

In section 8.1.9, remove blue formatting from this sentence:

..

This DVSEC capability contains one or more Register Block entries. Figure 131 illustrates a DVSEC Capability with 3 Register Block Entries.

..

In section 8.1.9.1, make the following changes:

Bit	Attributes	Description
...		

2:0	HWInit	... The Registers block must be wholly contained within the specified BAR. For a 64-bit Base Address Register, the Register BIR indicates the lower DWORD.
7:3	RsvdP	Reserved.
15:8	HwInit	Register Block Identifier - Identifies the type of CXL registers. Defined encodings are: • 00h Indicates the register block entry is empty and the Register BIR, Register Block Offset Low and Register Block Offset High fields are invalid. • 01h Component Registers. The format of the Component Register block is defined in Section 8.2.4. • 02h BAR Virtualization ACL Registers. The format of the BAR Virtualization ACL Register Block~~Component Register block~~ is defined in Section 8.2.7. • 03h CXL ~~Memory~~ Device Registers. The format of the CXL ~~Memory~~ Device Register block is defined in Section 8.2.8. All other Reserved.
31:16	HwInit	Register Block Offset Low - A[31:16] ~~of~~ byte offset from the starting address of the Function's BAR associated with the Register BIR field ~~address contained by one of the Function's Base Address Registers~~ to point to the base of the Register Block. Register Block Offset is 64K aligned. Hence A[15:0] is zero.
..	..	..

In section 8.1.9.2, make the following changes:

Bit	Attributes	Description
31:0	HwInit	Register Block Offset High - A[63:32] ~~of~~ byte offset from the starting address of the Function's BAR associated with the Register BIR field~~address contained by one of the Function's Base Address Registers~~ to point to the base of the Register Block.

In section 8.1.11.1, make the following change:

Data Object Byte Location	Length	Description
..	..	..
0Ah	2	EntryHandle - Handle value associated with the entry being requested. ~~EntryHandle=0 represents the very first entry in the table.~~ For Table Type = 0, EntryHandle = 0 specifies that the request is for the CDAT table header and EntryHandle=1-N indicates the request is for the CDAT Structure[EntryHandle – 1].

In section 8.2.5, make the following change:

Table 142 CXL_Capability_ID Assignment

Capability	ID	Highest version	Mandatory[1]	Not Permitted	Optional
..	..	..	..	..	..
CXL Extended Security Capability (Section 8.2.5.13)	6	1	R	All others	
..	..	..	..	..	..

1 P- PCI Express device, D1 - CXL 1.1 Device, D2 - CXL 2.0 Device, LD - Logical Device, FMLD - Fabric Manager owned LD 0xFFFF, UP1 - CXL 1.1 Upstream Port RCRB, DP1 - CXL 1.1 Downstream Port RCRB, R - CXL 2.0 Root Port (includes CXL Host Bridge registers), USP - CXL Switch Upstream Port, DSP - CXL Switch Downstream Port

In section 8.2.5.1, make the following change:

Bit Location	Attributes	Description
…		
19:16	RO	CXL_Capability_Version: This defines the version number of the CXL_Capability structure present. For this and the prior version of the specification, this field must be 0x1.
…		

In section 8.2.5.9.6, make the following change:

Header Log Registers are accessed as series of 32 bit wide individual registers even though it is represented as a single 512 bit long entity for convenience. In accordance with the section 8.2.2, each individual register shall be accessed as an aligned 4 Byte quantity.

Bit Location	Attributes	Description
511:0	RO	Header Log: The information logged here depends on the type of Uncorrectable Error Status bit recorded as described in Section 8.2.5.9.1. If multiple errors are logged in Uncorrectable Error Status register, First_Error_Pointer field in Error Capabilities and Control Register identifies the error that this log corresponds to.

In section 8.2.5.12, make the following change:

CXL HDM Decoder Capability Structure enables interleaving of HDM across CXL.mem-capable devices.

A CXL Host Bridge is identified as an ACPI device with Hardware~~Host Interface~~ ID (HID) of "ACPI0016" and is associated with one or more CXL Root ports. Any CXL 2.0 Host Bridge that is associated with more than one CXL Root Port must contain one instance of this capability structure in the CHBCR. This capability structure resolves the target CXL Root Ports for a given memory address.

In sections 8.2.5.12.7 and 8.2.5.12.15, make the following change:

Bit Location	Attributes	Description

..	..	..
12	RWL / RO	This bit is RWL for CXL Hosts and Upstream Switch Ports. This bit is RO for a CXL.mem device and it may return the value of 0 or 1. Target Device Type 0: Target is a CXL Type 2 Device 1: Target is a CXL Type 3 Device The locking behavior is described in Section 8.2.5.12.21. Default value of this field is 0.
..	..	..

Update section 8.2.5.14.3 as follows

Bit Location	Attributes	Description
3:0	RW1CS	Rx IDE Status: 0: Reserved 1: Active Containment Mode 2: Active Skid Mode 4: ~~Fail~~ Insecure State ~~Error~~ All other reserved
7:4		Tx IDE Status: 0: Reserved 1: Active Containment Mode 2: Active Skid Mode 4: ~~Fail~~ Insecure State ~~Error~~ All other reserved
..	..	..

Update section 8.2.5.14.4 as follows

Bit Location	Attributes	Description
3:0		Rx Error Status:

		Describes the error condition that transitioned the link to ~~Fail~~ Insecure ~~Mode~~ State .. 0b0010: MAC ~~Header~~ or Truncated MAC received when the link is not in secure mode (when integrity is not enabled and the receiver detects 0b0011: MAC header received when not expected (No MAC ~~EPCOH~~EPOCH running but the receiver detects a MAC header) 0b0100: MAC Header not received when expected (MAC header not received within 6 flit after MAC ~~EPCOH~~EPOCH has terminated) 0b0101: Truncated MAC flit received when not expected (if the receiver gets truncated MAC flit corresponding to a completed MAC ~~EPCOH~~EPOCH) MAC header) ..
..	..	..

In section 8.2.8, make the following changes:

CXL device registers are mapped in memory space allocated via a standard PCIe BAR. The entry in the Register Locator DVSEC structure (Section 8.1.9) with Register Identifier = 03h (CXL Device Registers Section 8.2.8) describes the BAR number and the offset within the BAR where these registers are mapped. The PCIe BAR shall be marked as prefetchable in the PCI header. At the beginning of the CXL device register block is a CXL Device Capabilities Array Register which defines the size of the CXL Device Capabilities Array followed by a list of CXL Device Capability headers. Each header contains an offset to the capability specific register structure from the start of the CXL device register block.

No registers defined in 8.2.8 are larger than 64-bits wide so that is the maximum access size allowed for these registers. If this rule is not followed, the behavior is undefined. To illustrate how the fields fit together, the layouts in section 8.2.8.1, section 8.2.8.2, and Figure 139. Mailbox Registers are shown as greater than 128-bit register. Implementations are expected to use any size accesses for this information up to 64-bits without loss of functionality – the information is designed to be accessed in chunks, each no greater than 64-bits.

Figure 138. CXL ~~Memory~~ Device Registers

...

In section 8.2.8.1, make the following changes:

Bits	Attributes	Description
...		
27:24	RO	Type: Identifies the type-specific capabilities in the CXL Device Capabilities Array.

		• 0h = No type-specific capabilities or type is inferred from the PCI class code. • 1h = Memory Device Capabilities (Section 8.2.8.5). Other values reserved.
31:284	RO	Reserved
…		
127:48	RO	Reserved

In section 8.2.8.2.1, make the following change:

CXL device capability register structures are identified by a 2-byte identifier ~~as specified in the table below~~.

- Capability identifiers 0000h-3FFFh describe generic CXL device capabilities as specified in the table below.
- Capability identifiers 4000h-7FFFh describe type-specific capabilities associated with the type specified in the CXL Device Capabilities Array Register (Section 8.2.8.1) ~~Class Code register in the PCI Header (Offset 09h)~~.
- Capability identifiers 8000h-FFFFh describe vendor specific capabilities.

Capability identifiers 0000h-3FFFh that are not specified in this table are reserved.

Capability ID	Description	Required*	Version
…			

*M = mandatory for all devices that ~~advertises~~implement the CXL Device Register entry (Register Block Identifier=03h) in the Register Locator DVSEC (Section 8.1.9); O = Optional.

In section 8.2.8.4, make the following changes:

…

In case of a timeout, the caller may attempt to recover the device by either issuing CXL or Conventional reset~~, hot reset, warm reset or a cold reset~~ to the device.

When a command is successfully started as a background operation, the device shall return the Background Command Started return code defined in Section 8.2.8.4.5.1. While the command is executing in the background, the device should update the percentage complete in the Background Command Status Register at least once per second. Once the command completes in the background, the device shall update the Background Command Status Register with the appropriate return code as defined in Section 8.2.8.4.5.1. The caller may then retrieve the results of the background operation ~~by issuing a new command~~ from the Background Command Status Register.

The mailbox registers are described below.

...

In section 8.2.8.4.3, make the following changes:

Bits	Attributes	Description
...		
18:11	RsvdP	Reserved
22:19	RO	Type: Identifies the type-specific commands supported by the mailbox. • 0h = No type-specific commands or type is inferred from the PCI class code. • 1h = Memory Device Commands (Section 8.2.9.5). Other values reserved.
31:2311	RsvdP	Reserved

In section 8.2.8.4.5, make the following change:

...

Bits	Attributes	Description
...		
36:16	RW	Payload Length: The size of the data in the command payload registers (0-Payload Size specified in the Mailbox Capabilities Register). Expressed in bytes. Written by the caller to provide the command input payload size to the device prior to setting the doorbell. Written by the device to provide the command output payload size to the caller when the doorbell is cleared.
...		

In section 8.2.8.4.5.1, make the following changes:

Table 150. Command Return Codes

Value	Definition
...	

000Ah	FW ~~Authentication~~ Verification Failed: The FW package was not saved to the device because the FW package ~~authentication~~ verification failed.
…	
000Eh	Invalid Handle: One or more Event Record Handles were invalid or specified out of order.
…	
0017h	Invalid Log: The log page is not supported or not valid.

In section 8.2.8.5, make the following changes:

8.2.8.5 Memory Device ~~Registers~~Capabilities

This section describes the capability registers specific to CXL memory devices that implement the PCI Header Class Code Register ~~as~~ defined in Section 8.1.12.1 or advertise Memory Device Capabilities support in the CXL Device Capabilities Array Register (Section 8.2.8.1).

~~CXL memory device register structures are identified by a 2-byte identifier as specified in the table below. Capability identifiers 4000h-7FFFh describe capabilities registers specific to CXL memory devices that implement the PCI Header Class Code Register as defined in Section 8.1.12.1.~~

CXL memory device capability ~~Capability~~ identifiers 4000h-7FFFh that are not specified in this table are reserved.

Table 151. CXL Memory Device Capabilities Identifiers

…			

*M = mandatory for all CXL memory devices ~~that implement a Register DVSEC Locator entry with Register Block Identifier=03h~~; PM = mandatory for CXL memory devices that support persistence ~~and implement a Register DVSEC Locator entry with Register Block Identifier=03h~~; O = Optional.

In section 8.2.8.4.7, make the following change:

Reports information about the last command executed in the background since the last ~~cold or warm or hot~~Conventional reset.

…

In section 8.2.8.5.1.1, make the following changes:

Bits	Attributes	Description
..	..	..

7:5	RO	Reset Needed: When non-zero, indicates the least impactful reset type needed to return the device to the operational state. A cold reset is considered more impactful than a warm reset. A warm reset is considered more impactful that a hot reset, which is more impactful than a CXL reset. This field returns non-zero value if FW Halt is set, ~~ or~~ Media Status is ~~not in the ready state~~ in the Error or Disabled state, or the Mailbox Interfaces Ready does not become set . • 000b =Device is operational, and no reset is required • 001b = Cold Reset • 010b = Warm Reset • 011b = Hot Reset • 100b = CXL Reset (Device must not report this value if it does not support CXL Reset) All other encodings are reserved.
...		

Throughout Chapter 8, make the following substitution:

Invalid ~~Parameter~~Input

In section 8.2.9, make the following changes:

CXL device commands are identified by a 2-byte Opcode ~~as specified in the table below~~.

- Opcodes 0000h-3FFFh describe generic CXL device cmmands as specified in the table below.
- Opcodes 4000h-BFFFh describe ~~Class Code specific commands~~type-specific commands associated with the type specified in the Mailbox Capabilities Register (8.2.8.4.3).
- Opcodes C000h-FFFFh describe vendor specific commands.

~~Opcodes 0000h-3FFFh that are not specified in this table are reserved.~~

Opcodes also provide an implicit ... by adding backward-compatible changes.

Opcodes 0000h-3FFFh that are not specified in this table are reserved.

Table 152. CXL Device Command Opcodes

...			

*M = mandatory for all devices that implement ~~a Register Locator entry with Register Block Identifier =03h~~the CXL Device Register entry (Identifier=03h) in the Register Locator DVSEC (Section 8.1.9); O = Optional.

...

In section 8.2.9.1.1, make the following change:

...

A ~~device implement a Register DVSEC Locator entry with Register Block Identifier=03h~~ CXL memory device that implements the PCI Header Class Code defined in Section 8.1.12.1 or advertises Memory Device Command support in the Mailbox Capabilities Register (Section 8.2.8.4.3) shall utilize the Memory Module Event Record format when reporting general device events and shall utilize either the General Media Event Record or DRAM Event Record when reporting media events.

...

In section 8.2.9.1.2, add the following:

...

Possible Command Return Codes:

- ...
- Invalid Payload Length
- Media Disabled
- Busy

...

In section 8.2.9.1.3, make the following changes:

...

If the Event Log has overflowed, the host may clear all the device's stored event logs for the requested Event Log instead of explicitly clearing each event with the unique handle. Events shall be cleared in temporal order. The device shall verify the event record handles specified in the input payload are in temporal order. If the device detects an older event record that will not be cleared when Clear Event Records is executed, the device shall return the Invalid Handle return code and shall not clear any of the specified event records.

Possible Command Return Codes:

- ...
- Invalid Payload Length
- Media Disabled
- Busy

...

Table 160. Clear Event Records Input Payload

Byte Offset	Length	Description
...		
1	1	Clear Event Flags: If 0, the device shall clear the events records specified in the Event Record Handles list. • Bit[0]: Clear All Events: When set, the device shall clear all events that it currently has stored internally for the requested Event Log. ~~When utilizing this mechanism, if the event log has not~~

		~~overflowed, it is possible to clear events that the host has not yet been notified of.~~Clear All Events is only allowed when the Event Log has overflowed, otherwise the device shall return Invalid Input. • Bits[7:1]: Reserved
…		

…

In section 8.2.9.1.4, add the following:

Retrieve the settings for interrupts that are signaled for device events.

Possible Command Return Codes:

…

In section 8.2.9.1.5, add the following:

Change the settings for the interrupts that are signaled for device events. All event log interrupt settings shall be reset to 00b (No Interrupts) by the device on Conventional reset.

…

In section 8.2.9.2, add the following:

The number of FW slots the device supports is vendor specific, up to four. The minimum FW slots supported shall be two, one slot for the active FW and one slot for a FW package that is staged for activation. Only one slot may be active at a time and only one slot may be staged for activation at a time. The staged FW will transition to the active FW on power cycle. Other FW packages that are fully transferred and stored in a slot persist across power cycles.

…

In section 8.2.9.2.2, make the following changes:

..

If the FW package is transferred in its entirety, the caller makes one call to Update FW with Action = Full FW Transfer.

If a FW package is transferred in parts, the caller makes one call to Transfer FW with Action = ~~Start~~Initiate FW Transfer, zero or more calls with Action = Continue FW Transfer, and one call with Action = ~~Finish~~ End Transfer or Abort Transfer. The FW package parts shall be transferred in order, otherwise the device shall return the FW Transfer Out of Order return code.

If the FW package transfer is interrupted by a Conventional or CXL reset, the FW package transfer shall be aborted by the device. If a FW package transfer is aborted prior to the entire FW package being successfully stored on the device, the device shall require the FW package transfer to be started from the beginning of the FW package.

…

Possible Command Return Codes:

- ...
- FW ~~Authentication~~ Verification Failed
- ...

In section 8.2.9.3.1, make the following changes:

...

Table 166. Get Timestamp Output Payload

Byte Offset	Length	Description
0	8	Timestamp: The current device timestamp which represents the value set with the Set Timestamp command plus the number of ~~unsigned~~ nanoseconds that have elapsed since ~~midnight, 01-Jan-1970, UTC~~ the timestamp was set.

...

In section 8.2.9.3.2, make the following change:

Set the timestamp on the device. It is recommended that the host set the timestamp after every ~~hot reset, every warm reset, every cold reset, and every function level~~ Conventional or CXL reset. Otherwise, the timestamp may be inaccurate.

...

In section 8.2.9.4.1, add the following:

...

Table 169. Get Supported Logs Supported Log Entry

Byte Offset	Length	Description
...		
10h	4	Log Size: The maximum number of bytes of log data available to retrieve for the log identifier.

In section 8.2.9.4.2, add the following:

...

Possible Command Return Codes:

- ..
- Invalid Payload Length

- Invalid Log
- Media Disabled
- Busy

...

In section 8.2.9.5, make the following changes:

~~CXL memory device commands are identified by a 2-byte Opcode as specified in the table below. Opcodes 4000h-BFFFh describe CXL memory device specific commands.~~

~~Opcodes 4000-BFFFh that are not specified in this table are reserved.~~

This section describes the commands specific to CXL memory devices that implement the PCI Header Class Code defined in Section 8.1.12.1 or advertise Memory Device Command support in the Mailbox Capabilities Register (Section 8.2.8.4.3).

Opcodes also provide an implicit ... by adding backward-compatible changes.

Opcodes 4000-BFFFh that are not specified in this table are reserved.

Table 174. CXL Memory Device Command Opcodes

Opcode					Required	Optional	Input Payload Size (B)	Output Payload Size (B)
Command Set Bits[15:8]		Command Bits[7:0]		Combined Opcode				
41h	Capacity Config and Label Storage	...						
		01h	Set Partition Info (Section 8.2.9.5.2.2)	4101h	O		~~0Ah~~9	0
		...						
...								

*M = mandatory for all CXL memory devices ~~that implement a Register DVSEC Locator entry with Register Block Identifier=03h~~; PM = mandatory for CXL memory devices that support persistence ~~and implement a Register DVSEC Locator entry with Register Block Identifier=03h~~; O = Optional.

...

In section 8.2.9.5.1.1, make the following change:

Table 175. Identify Memory Device Output Payload

Byte Offset	Length	Description

…		
38h	4	LSA Size: The size of the Label Storage Area. Expressed in bytes. The minimum LSA size is defined in Section 9.13.2.
…		
41h	1	Poison Handling Capabilities: The device's poison handling capabilities. • Bit[0]: Injects Persistent Poison – When set and the device supports poison injection, any poison injected in nonvolatile DPA address shall remain persistent across all types of device resets. When clear and the device supports poison injection, ~~hot reset, warm reset,~~ Conventional or CXL reset ~~or cold reset~~ shall clear the injected poison automatically. • …
…		

In section 8.2.9.5.2.2, make the following changes:

Set the partitioning between volatile capacity and persistent capacity for the partitionable capacity. Partitionable capacity is equal to (Total Capacity - Volatile Only Capacity - Persistent Only Capacity). This command shall fail with an Unsupported error if there is no partitionable capacity (i.e. Identify Memory Device reports Partition Alignment as zero). The device shall return Invalid Input if the specified capacity is not aligned to the partition alignment requirement reported in the Identify Memory Device command. Using this command to change the size of the persistent capacity shall result in the loss of data stored.

..

- …
- Invalid Payload Length
- Invalid Security State
- Media Disabled
- Busy

Table 177. Set Partition Info Input Payload

Byte Offset	Length	Description
0	8	Volatile Capacity: The amount of partitionable capacity that shall be allocated to volatile capacity, in multiples ~~in~~of 256 MB aligned to the partition alignment requirement reported in the Identify Memory Device command. The remainder of the partitionable capacity shall be allocated to persistent capacity.
..	..	..

In section 8.2.9.5.2.3, make the following changes:

... The format of the LSA is specified in Section 9.13~~4~~.2. The size of the Label Storage Area is retrieved from the Identify Memory Device command.

Possible Command Return Codes:

- ...
- ~~Invalid Security State~~
- Invalid Payload Length
- Busy

...

In section 8.2.9.5.2.4, make the following changes:

The format of the Label Storage Area is specified in Section 9.14~~4~~3.2.

Possible Command Return Codes:

- ...
- ~~Invalid Security State~~
- Invalid Payload Length
- Busy

...

In section 8.2.9.5.3.1, remove blue formatting from table 181

In section 8.2.9.5.3.2, make the following changes:

Retrieve the device's critical alert and programmable warning configuration. Critical alerts shall automatically be configured by the device after a ~~device~~ Conventional reset. If supported, programmable warning thresholds shall be initialized to vendor recommended defaults by the device ~~up~~on ~~device~~ Conventional reset.

...

Table 182. Get Alert Configuration Output Payload

Byte Offset	Length	Description
...		
1	1	.. Bit[1]: When set, the Device Over-Temperature Programmable Warning Threshold field is programmable by the host

		Bit[~~1~~2]: When set, the Device Under-Temperature Programmable Warning Threshold field is programmable by the host Bit[3]: When set, the Corrected Volatile Memory Error Programmable Warning is programmable by the host ..
...		

In section 8.2.9.5.3.3, make the following changes:

Set Alert Configuration allows the host to optionally configure programmable warning thresholds. If supported, programmable warning thresholds shall be initialized to vendor recommended defaults by the device ~~up~~on ~~device~~ Conventional reset. After completion of this command the requested programmable warning thresholds shall replace any previously programmed warning thresholds.

...

In section 8.2.9.5.3.4, add the following:

...

Possible Command Return Codes:

- ...
- Invalid Payload Length
- Media Disabled

...

In section 8.2.9.5.3.5, add the following:

Possible Command Return Codes:

- ...
- Invalid Payload Length
- Media Disabled

...

Table 185. Set Shutdown State Input Payload

Byte Offset	Length	Description
0	1	State: The current shutdown state • Bit[0]: Dirty – A one value sets the device's internal shutdown state to "dirty", a zero value sets it to "clean". The device shall persistently store this state and use it after the next Conventional reset~~, hot, warm or cold~~, to determine if the Dirty Shutdown Count described in Section

		• ...

In section 8.2.9.5.4.4, make the following correction:

Table 191. Get Scan Media Capabilities Input Payload

Byte Offset	Length	Description
0	8	Get Scan Media Capabilities Start Physical Address: The starting DPA from where to retrieve Scan Media capabilities. • Bits[5:~~2~~0]: Reserved • ...
...		

In section 8.2.9.5.4.5, make the following correction:

Table 193. Scan Media Input Payload

Byte Offset	Length	Description
0	8	Scan Media Physical Address: The starting DPA where to start the scan. • Bits[5:~~2~~0]: Reserved • ...
...		

In section 8.2.9.5.6, make the following change:

Persistent Memory security is an optional feature that gates access to persistent memory with a user passphrase. When enabled, the persistent memory shall be locked on a ~~hot, warm or cold~~Conventional reset until the user passphrase is supplied with the Unlock command. When the persistent memory is locked, any commands that require access to the media shall return the Invalid Security State return code.

...

In section 8.2.9.5.6.2, make the following changes:

Set or change the user or master passphrase. When the user passphrase is set, the device persistent memory shall be locked on ~~hot, warm or cold~~Conventional reset until the user passphrase is supplied.

...

Possible Command Return Codes:

- ...
- Invalid Payload Length
- Busy

...

In section 8.2.9.5.6.3, make the following changes:

Disable the user or master passphrase. When the user passphrase is disabled, the device persistent memory shall not be locked on ~~hot, warm or cold~~Conventional reset. When the master passphrase is disabled, the device shall return Invalid Input for the Passphrase Secure Erase command with the master passphrase.

Possible Command Return Codes:

- ...
- Invalid Payload Length
- Busy

...

In section 8.2.9.5.6.4, add the following:

...

Possible Command Return Codes:

- ...
- Invalid Payload Length
- Busy

...

In section 8.2.9.5.6.6, make the following changes:

...

Possible Command Return Codes:

- ...
- Invalid Payload Length
- Busy

...

Table 199. Passphrase Secure Erase Input Payload

Byte Offset	Length	Description
..	..	..
20h	20h	Current Passphrase: The current passphrase. If Passphrase Type is 01h (User Passphrase) ~~Ignored~~ and ~~if~~ the user passphrase is not currently

		set or is not supported by the device, this value is ignored. If Passphrase Type is 00h (Master Passphrase), the master passphrase is required.

In section 8.2.9.5.7.1, add the following:

...

Possible Command Return Codes:

- ...
- Invalid Payload Length
- Busy

...

In section 8.2.9.5.7.2, make the following changes:

...

Each Security Receive command returns the appropriate data corresponding to a Security Send command as defined by the rules of the Security Protocol. The Security Receive command data may or may not be ~~retaining~~ retained if there is a loss of communication between the device and the host, or if a ~~device hot, warm or cold~~Conventional reset occurs.

...

F5 Chapter 9 Miscellaneous errata

In section 9.12.3, make the following correction:

...

2. ... System Firmware may leave the Unmask SBR and the Unmask Link Disable bits in Port Control ~~Override~~ register of the Downstream Port at the default (0) values to prevent legacy PCIe software from resetting the device and the link respectively.

...

In section 9.13.1, make the following change:

...

- CXL 2.0 Type 3 devices must support at least one of the two IG groups as reported via HDM Decoder Capability Register (Section 8.2.5.12.1).

— Group 1: Interleaving on HPA[8], HPA[9], ~~and~~ HPA[10] and HPA[11]
— Group 2: Interleaving on ~~HPA[11],~~ HPA[12], HPA[13] and HPA[14]

In section 9.13.2, make the following change:

...

In addition, the ~~Get LSA Size~~Identify Memory Device mailbox command exposes the size of the LSA for a given CXL memory device.

...

In section 9.13.2.2, make the following change:

...

Table 210. Label Index Block Layout

Byte Offset	Length	Description
...		
48h	~~Varies~~B7h	Free: NSlot bits, padded with zeros to align index block to 256 bytes.

...

In Figures 155 and 156, replace the term "HDM Decoder DVSEC" with "HDM Decoder".

In section 9.14.1.1, make the following change:

Table 214. CEDT Header

Field	Byte Length	Byte Offset	Description
Header			
Signature	4	0	'CEDT'. Signature for the CXL Early Discovery Table.
Length	4	4	Length, in bytes, of the entire CEDT.
Revision	1~~2~~	8	1
..	..	..	..

F6 Secure MEM tests

In sections 14.11.5.4 and 14.11.5.5, please make the following changes:

14.11.5.4: CXL.mem Poison injection.

Required Device Capabilities:
• The CXL device must support ~~Algorithm 1a, and~~ Link Layer Error Injection Capabilities.

Test Steps:

~~**1. Setup device for Multiple Write Streaming:**~~

~~**a. Write a pattern {64{8'hFF}} to cache aligned address A1**~~

~~**b. Write a Compliance mode DOE to inject Poison**~~

1. Select a memory target range on the Device Physical Address (DPA) that belongs to the Device Under Test
2. Translate the DPA to a Host Physical Address (HPA)
3. Perform Continuous read/write operations on the HPA
4. Write a compliance mode DOE to inject Poison errors.

Table 237. Mem-Poison Injection Request

Data Object Byte Offest	Length	Description	Value
0h	8	Standard DOE Request Header	
8h	1	Request Code	6, Poison Injection
9h	1	Version	2
0Ah	2	Reserved	
0Ch	1	Protocol	3

~~a. Write Compliance mode DOE with the following request:~~

~~**Table 238. Multie Write Streaming Request**~~

~~5. Data Object Byte Offest~~	~~6. Length~~	~~7. Description~~	~~8. Value~~

~~0h~~	~~8~~	~~Standard DOE Request Header~~	
~~8h~~	~~1~~	~~Request Code~~	~~3, Multiple Wrtie Streaming~~
~~9h~~	~~1~~	~~Version~~	~~2~~
~~0Ah~~	~~2~~	~~Reserved~~	
~~0Ch~~	~~1~~	~~Protocol~~	~~3~~
~~0Dh~~	~~1~~	~~Virtual Address~~	~~0~~
~~0Eh~~	~~1~~	~~Self-Checking~~	~~0~~
~~0Fh~~	~~1~~	~~Verify Read Semantics~~	~~0~~
~~10h~~	~~1~~	~~Num Increments~~	~~0~~
~~11h~~	~~1~~	~~Num Sets~~	~~0~~
~~12h~~	~~1~~	~~Num Loops~~	~~1~~
~~13h~~		~~Reserved~~	
~~14h~~	~~8~~	~~Start Address~~	~~A1~~
~~1Ch~~	~~8~~	~~Write Address~~	~~0~~
~~24h~~	~~8~~	~~Write Back Address~~	~~A2 (Must be distinct from A1)~~
~~2Ch~~	~~8~~	~~Byte Mask~~	~~0xFFFFFFFFFFFFFFFF~~
~~34h~~	~~4~~	~~Address Increment =~~	~~0~~
~~38h~~	~~4~~	~~Set Offset~~	~~0~~
~~3Ch~~	~~4~~	~~Pattern "P"~~	~~0xAA~~
~~40h~~	~~4~~	~~Increment Pattern "B"~~	~~0~~

...

14.11.5.5: CXL.mem CRC injection.

Required Device Capabilities:
• The CXL device must support ~~Algorithm 1a, and~~ Link Layer Error Injection Capabilities.

Test Steps:

~~1. Setup device for Multiple Write Streaming:~~

~~a. Write a pattern {64{8'hFF}} to cache aligned address A1~~

~~b. Write a Compliance mode DOE to inject Poison~~

1. Select a memory target range on the Device Physical Address (DPA) that belongs to the Device Under Test
2. Translate the DPA to a Host Physical Address (HPA)
3. Perform Continuous read/write operations on the HPA
4. Write a compliance mode DOE to inject Poison errors.

Table 239. Mem CRC Injection Request

Data Object Byte Offest	Length	Description	Value
0h	8	Standard DOE Request Header	
8h	1	Request Code	7, CRC Injection
9h	1	Version	2
0Ah	2	Reserved	
0Ch	1	Protocol	3
		Num Bits Flipped	1
		Num Flits Injected	1

~~a. Write Compliance mode DOE with the following request:~~

~~**Table 240. Multic Write Streaming Request**~~

~~5. Data Object Byte Offest~~	~~6. Length~~	~~7. Description~~	~~8. Value~~
~~0h~~	~~8~~	~~Standard DOE Request Header~~	
~~8h~~	~~1~~	~~Request Code~~	~~3, Multiple Write Streaming~~
~~9h~~	~~1~~	~~Version~~	~~2~~
~~0Ah~~	~~2~~	~~Reserved~~	
~~0Ch~~	~~1~~	~~Protocol~~	~~3~~
~~0Dh~~	~~1~~	~~Virtual Address~~	~~0~~

~~0Eh~~	~~1~~	~~Self-Checking~~	~~0~~
~~0Fh~~	~~1~~	~~Verify Read Semantics~~	~~0~~
~~10h~~	~~1~~	~~Num Increments~~	~~0~~
~~11h~~	~~1~~	~~Num Sets~~	~~0~~
~~12h~~	~~1~~	~~Num Loops~~	~~1~~
~~13h~~		~~Reserved~~	
~~14h~~	~~8~~	~~Start Address~~	~~A1~~
~~1Ch~~	~~8~~	~~Write Address~~	~~0~~
~~24h~~	~~8~~	~~Write Back Address~~	~~A2 (Must be distinct from A1)~~
~~2Ch~~	~~8~~	~~Byte Mask~~	~~0xFFFFFFFFFFFFFFFF~~
~~34h~~	~~4~~	~~Address Increment =~~	~~0~~
~~38h~~	~~4~~	~~Set Offset~~	~~0~~
~~3Ch~~	~~4~~	~~Pattern "P"~~	~~0xAA~~
~~40h~~	~~4~~	~~Increment Pattern "B"~~	~~0~~

...

F7 Compliance DVSEC length incorrect

This Erratum should apply to both CXL 1.1 and CXL 2.0

CXL Device Test Capability Advertisement length is incorrectly listed as 22h.

Table 256. DVSEC Registers

...			
Designated Vendor Specific Header 1 (offset 04h)	31:20	DVSEC Length	~~22h~~ 1Ch
...			

F8 Flag Bit in Algorithm 2 not documented

This Erratum should apply to both CXL 1.1. and CXL2.0

In section 14.3.5 make the following change:

This Algorithm aims to test the scenario where a Device is a producer and the CPU is a consumer. Device simply executes a pre-determined Algorithm of writing known patterns to a data location followed by a flag update write. Threads on the CPU poll the flag followed by reading the data patterns, followed by polling the flag again. This is a simple way of making sure the required ordering rules of producer consumer workloads are being followed through the stack. Device only participates in the execute phase of this Algorithm. Figure 194 illustrates the device execute phase. The Verify phase is run on the CPU, software reads addresses in the following order [F, X, (X+Y)…(X+N*Y), F]. Knowing the value of the flag at two ends, the checker knows the range in which [X, (X+Y)…(X+N*Y)] have to be in. For example, if P=0, the first read of F returns a value of 3 and the next read of F returns a value of 4, then checker knows that all intermediate values have to be either 3 or 4. Moreover, if the device is using strongly ordered semantics, then the checker should never see a transition of values from 3 to 4 (implying monotonically decreasing values for the non-flag addresses). If using CXL.cache protocol, device must ensure global observability of previous "data" writes before updating the flag. When using strongly ordered semantics, each update must be globally visible before the next one. Depending on the flow used for dirty evicts, this can be implementation specific. It is the responsibility of the device to ensure that the writes in the execute phase are globally observable before updating the flag "F". The "PatternParameter" field is not relevant for this Algorithm. The Flag "F" should be written to Register 2: "WriteBAckADdress1" in the Device Capabilities to support the Test Agorithms.

F9 Completion Timeout Injection missing pass criteria

Please add the following to section 14.11.5.8:

...

Table 246. Multie-Write Streaming Request

Data Object Byte Offset	Length	Description	Value
11h	1	Num Sets	0
12h	1	Num Loops	1
13h		Reserved	
14h	8	Start Address	A1
1Ch	8	Write Address	0
24h	8	WriteBackAddress	A2 (Must be distinct from A1)
2Ch	8	Byte Mask	0xFFFFFFFFFFFFFFFF
34h	4	Address Increment =	0
38h	4	Set offset	0
3Ch	4	Pattern "P"	0xAA
40h	4	Increment Pattern "B"	0

Pass Criteria:

- CXL.cache IDE link state remains secure
- Host Reciever logs link error

...

F10 Removal of text in 14.11.5

In section 14.11.5, please remove the following:

14.11.4 Security RAS

~~Make these tests pointers back to current RAS tests. Pass criteria needs a comment that the link remains in a secure state.~~

~~CRC Injections should work wihtout a Protocol Analyzer, since it has been added as an injection hook to the Compliance DOE~~

...

F11 Compliance DOE referencing incorrect table

In section 14.16.4.1 please make the following changes

Table 278. Compliance Mode Availability Response

Data Object Byte Offset	Length	Description
..	..	..
08h	1	Status: See table ~~192~~ 276 for error codes
..	..	..

In section 14.16.4.3 please make the following changes

Table 282. Compliance Mode Halt All Response

Data Object Byte Offset	Length	Description
..	..	..
08h	1	Status: See table ~~192~~ 276 for error codes

In section 14.16.4.4 please make the following changes

Table 284. Compliance Mode Multiple Write Streaming Response

Data Object Byte Offset	Length	Description
..	..	..
08h	1	Status: See table ~~192~~ 276 for error codes

In section 14.16.4.5 please make the following changes

Table 286. Compliance Mode Producer Consumer Response

Data Object Byte Offset	Length	Description
..	..	..
08h	1	Status: See table ~~192~~ 276 for error codes

In section 14.16.4.6 please make the following changes

Table 288. Inject Bogus Writes Response

Data Object Byte Offset	Length	Description
..	..	..
08h	1	Status: See table ~~192~~ 276 for error codes

In section 14.16.4.7 please make the following changes

Table 290. Poison Inject Response

Data Object Byte Offset	Length	Description
..	..	..
08h	1	Status: See table ~~192~~ 276 for error codes

In section 14.16.4.8 please make the following changes

Table 292. Compliance Mode Availability Response

Data Object Byte Offset	Length	Description
..	..	..
08h	1	Status: See table ~~192~~ 276 for error codes

In section 14.16.4.9 please make the following changes

Table 294. Flow Control Injection Response

Data Object Byte Offset	Length	Description
..	..	..
08h	1	Status: See table ~~192~~ 276 for error codes

In section 14.16.4.10 please make the following changes

Table 296. Cache Flush Injection Response

Data Object Byte Offset	Length	Description
..	..	..
08h	1	Status: See table ~~192~~ 276 for error codes

F12 Compliance Algorithm Security

In section 14.16.2 make the following changes:

Table 268. Bit 32 Expand description, and change Attribute to RW/RO

32	RW/RO	AddressIsVirtual: If set to 1, indicates that all programmed addresses are virtual and need to be translated by the device (via ATS). Useful for testing virtualization/device TLBs If the device only supports Virtual Address for security reasons, this bit is set to 1 and is Read Only. If device only support Physical addresses, bit is set to 0 and is Read Only.

In section 14.16.4.4 Add the following text following Table 284

If the device only supports Virtual Addresses, and the Virtual Address is set to 0, the return value shall be 0x01 "Not Authorized" If the device only supports Physical Addresses, and the Virtual Address is set to 1, the return value shall be 0x01 "Not Authorized"

In section 14.16.4.5 Add the following text following Table 286

If the device only supports Virtual Addresses, and the Virtual Address is set to 0, the return value shall be 0x01 "Not Authorized" If the device only supports Physical Addresses, and the Virtual Address is set to 1, the return value shall be 0x01 "Not Authorized"

F13 Update to 14.8.5 to remove secondary verify statement

In section 14.8.5 DVSEC CXL Capability, please remove the following verify statement:

~~4. Verify:~~

~~Variable~~	~~Value~~	~~Condition~~
~~R1~~	~~= R2~~	~~CONFIG_LOCK = 1~~
~~R2~~	~~!= R2~~	~~CONFIG_LOCK = 0~~

F14 Removal of Reference to CXL_ERROR_DOE_MAILBOX

In section 14.8.3 please make the following changes

Remove Test step 3

…

~~3. Verify:~~

~~a. CXL_ERROR_DOE_MAILBOX contains only one valid Data Object Protocol~~

Update the Pass Critera to state:

Pass Criteria

- Test 14.8.2 Passed
- ~~Verify Conditions Met~~ Either Compliance or CDAT DOE mailbox has a valid response.

F15 Incorrect CXL Capability Version

In Section 14.13.2, in the verify section, please update the following:

5. Verify:

Variable	Value	Condition
CXL_Capability_ID	02h	Always
CXL_Capbility_Version	0~~1~~2h	Always

F16 Updates to Memory Mapped Register Test

In section 14.13.13 through 14.13.16, the test steps need to be updated to reflect checking against the Device Capabilitys Array Register.

In section 14.13.13, please make the following change:

Test Steps:

1. Read Offset Primary_Mailbox_Register_Capabilities_Header_Base Length 4 bytes. Primary_Mailbox_Registers_Capabilities_Header_Base is obtained in test Section ~~14.13.11, "CXL Snoop Filter Capability Structure".~~ 14.13.12, "CXL Device Capabilities Array Register".

In section 14.13.14, please make the following change:

Test Steps:

1. Read Offset Primary_Mailbox_Register_Capabilities_Header_Base Length 4 bytes. Primary_Mailbox_Registers_Capabilities_Header_Base is obtained in test Section ~~14.13.11, "CXL Snoop Filter Capability Structure".~~ 14.13.12, "CXL Device Capabilities Array Register".

In section 14.13.15, please make the following change:

Test Steps:

1. Read Offset Primary_Mailbox_Register_Capabilities_Header_Base Length 4 bytes. Primary_Mailbox_Registers_Capabilities_Header_Base is obtained in test Section ~~14.13.11, "CXL Snoop Filter Capability Structure".~~ 14.13.12, "CXL Device Capabilities Array Register".

In section 14.13.16, please make the following change:

Test Steps:

1. Read Offset Primary_Mailbox_Register_Capabilities_Header_Base Length 4 bytes. Primary_Mailbox_Registers_Capabilities_Header_Base is obtained in test Section ~~14.13.11, "CXL Snoop Filter Capability Structure".~~ 14.13.12, "CXL Device Capabilities Array Register".

F17 Update to DVSEC CXL Lock test pass criteria

Seciton 14.8.7 'DVSEC CXL Lock" has the incorrect pass criteria.

In section 14.8.7 pass criteria, please make the following change:

Pass Criteria:

- Test ~~15.6.4~~ 14.8.4 Passed
- Verify Conditions Met

…

F18 Incorrect Pass criteria in 14.15.1 Sticky Register Tests

In section 14.15.1, please update the pass criteria with the following:

Pass Criteria:

- Test ~~15.6.2~~ 14.8.2 Passed

…

F19 Compliance Capability Return Value

In section 14.16.4.1 make the Add the following text following Table 278

The Available Capabilities and Enabled Capabilities bitmask values correspond to the request codes of each capability. Eg. The bit 1 will be set if the DOE supports the Request code 1, "Status", and bit 3 will be set if the DOE supports Request code 3, "Multiple Write Streaming"

F20 LinkError clarification

In section 5.1, add the following text to Table 58.

Table 58. Virtual LSM States Maintained Per Link Layer Interface

Virtual LSM State	Description
Reset	Power-on default state during which initialization occurs
Active	Normal operational state
L1.0	Power savings state, from which the link can enter Active via Retrain (maps to PCIe L1)
L1.1	Power savings state, from which the link can enter Active via Retrain (reserved for future use)
L1.2	Power savings state, from which the link can enter Active via Retrain (reserved for future use)
L1.3	Power savings state, from which the link can enter Active via Retrain (reserved for future use)
DAPM	Deepest Allowable PM State (not a resolved state; a request that resolves to an L1 substate)
SLEEP_L2	Power savings state, from which the link must go through Reset to reach Active
LinkReset	Reset propagation state resulting from software or hardware initiated reset

LinkError	Link Error state due to hardware detected errors which cannot be corrected through link recovery (eg. uncorrectable internal errors or surprise link down)
LinkDisable	Software controlled link disable state
Retrain	Transitory state that transitions to Active

Note: When the Physical Layer enters Hot-Reset or LinkDisable state, that state is communicated to all link layers as LinkReset or LinkDisable respectively. No ALMPs are exchanged, irrespective of who requested, for these transitions. LinkError must~~should~~ take the LTSSM to Detect or Disabled. For example, it is permitted to map CXL.io Downstream Port Containment to LinkError (when LTSSM is in Disabled state).

F21 PM Retry and Abort clarifications

In Section 5.1.2.5.1, modify the text as follows.

A State Status ALMP is sent after a valid State Request ALMP is received for ~~entry into~~ Active State (if the current vLSM state is already in Active, or if the current vLSM state is not Active and the request is following the entry into Active protocol) or PM States (when entry to the PM state is accepted). No State Status ALMP is sent if the PM state is not accepted.

In Section 5.1.2.4.2, add the following after Figure 88.

If PM entry is not accepted by the Downstream Port, it will not respond to the PM State Request. In this scenario:

- The Upstream Port is permitted to retry entry into PM with another PM State Request after a 1ms (not including time spent in recovery states) timeout, when waiting for a response for a PM State Request. Upstream Port must not expect a PM State Status response for every PM State Request ALMP. Even if it has sent multiple PM State Requests because of PM retries, if it receives a single PM State Status ALMP, it must move the corresponding vLSM to the PM state indicated in the ALMP. For a Downstream Port, if the vLSM is Active and it has received multiple PM State Request ALMPs for that vLSM, it is permitted to treat it as a single PM request and only respond with a single PM State Status if the vLSM transitions into the PM state. Figure 88a shows an example of this flow.
- The Upstream Port is also permitted to abort entry into PM by sending an Active State Request ALMP for the corresponding vLSM. Two scenarios are possible in this case:

a. Downstream Port receives the Active State Request before commit point of PM acceptance. It must abort PM entry and respond with Active State Status ALMP. Upstream port can begin flit transfer towards Downstream Port once it receives Active State Status ALMP. Since the vLSMs are already in Active state and flit transfer was already allowed from Downstream Port to Upstream Port direction during this flow, there is no Active State Request ALMP from Downstream Port to Upstream Port direction. Figure 88b shows an example of this flow.

b. Downstream Port receives the Active State Request after commit point of PM acceptance or after its vLSM is in PM state. Downstream Port must finish PM entry and send PM State Status ALMP (if not done so already). Upstream Port must treat the received PM State Status ALMP as an unexpected ALMP and trigger link Recovery. Figure 88c shows an example of this flow.

Figure 88a. Successful PM Entry following PM Retry

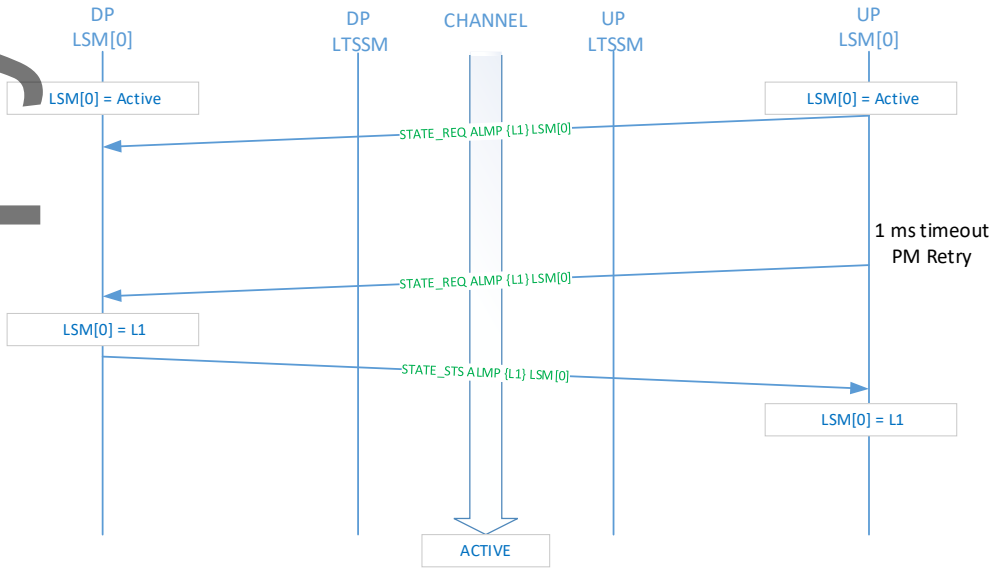


Figure 88b. PM Abort before Downstream Port PM acceptance

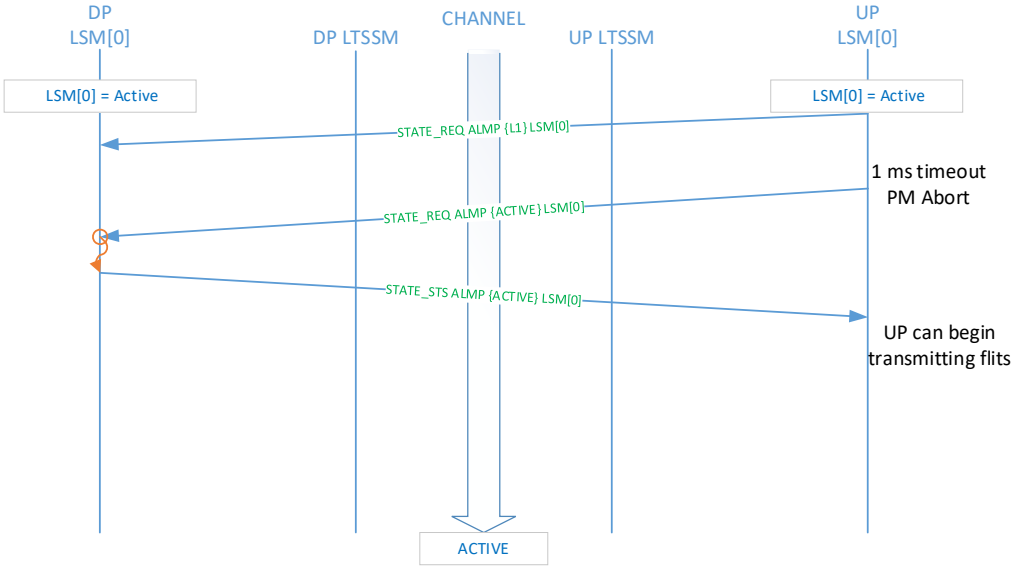
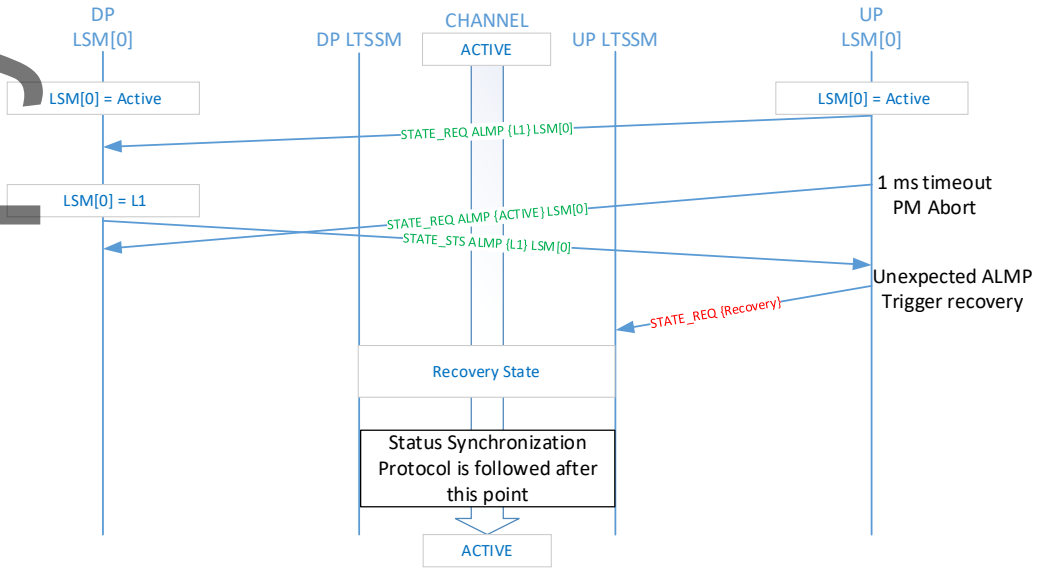


Figure 88c. PM Abort After Downstream Port PM acceptance



F22 PM Entry Phase 3 clarification

In Section 10.3.3, add the following text:

The third Phase is a conditional phase of PM entry and is executed only when all the Protocol interfaces of ARB/MUX have entered the same virtual PM state. The phase consists of bringing the Tx lanes to electrical ~~I~~idle and is always initiated by the Downstream Component. If the link transitions to recovery during or after entry into electrical idle, the Downstream Component must wait for at least 1us after reaching L0 before re-initiating entry into electrical idle. This is to allow enough time for an Active State Request ALMP transfer to occur in case either side wants to initiate a PM exit (and to give time for the remote ARB/MUX to stop requesting PM entry to LogPHY).

F23 Modifications to the CRC Error Injection within compliance

In Section 14.4.2 please make the following changes

Test Steps

1. Setup is the same as Test 14.3.6.1.2
2. While test is running, software will ~~repeat the~~ insert the following error injection. The Protocol Exerciser will retry the flit for at least MAX_NUM_RETRY times upon detecting a CRC error**:**

…

In Section 14.4.3 please make the following changes

Test Steps

1. Setup is the same as Test 14.3.6.1.2

2. While test is running, software will ~~repeat the~~ insert the following error injection. The Protocol Exerciser will retry the flit for at least MAX_NUM_RETRY x MAX_NUM_PHY_REINIT times upon detecting a CRC error:

…

F24 Querying Critical Component Information and Status

Update section 7.6.6.1 as follows

As the CXL system initializes, the FM can begin discovering all direct-attached CXL devices across all supported media interfaces. Devices supporting the FM API may be discovered using transport specific mechanisms such as the MCTP discovery process, as defined in the MCTP Base Specification. When a component is discovered, the FM shall issue the Identify command prior to issuing any other commands to check the component's type and its maximum supported command message size. A return of "Retry Required" indicates that the component is not yet ready to accept commands. After receiving a successful response to the Identify request, the FM may issue the Set Response Message Limit command to limit the size of response messages from the component based on the size of the FM's receive buffer. The FM shall not issue any commands with input arguments such that the command's response message exceeds the FM's maximum supported message size. Finally, the FM may issue~~s~~ Get ~~Supported~~ Log~~s~~, as defined in Section 8.2.9.4.2~~1~~, to read the Command Effects Log to determine which command opcodes are supported.

Update section 7.6.7.1.3 as follows

Table 89. Identify Switch Device Response Payload

Byte~~s~~ Offset	Length	Description
~~0~~		~~Device Management Version: Version of FM API command set supported by device. Currently 1.~~
~~1~~		~~Reserved~~
~~3-2~~		~~PCIe Vendor ID: As defined in PCIe 5.0 Base Specification~~
~~5-4~~		~~PCIe System ID: As defined in PCIe 5.0 Base Specification~~
~~7-6~~		~~PCIe Subsystem Vendor ID: As defined in PCIe 5.0 Base Specification~~
~~9-8~~		~~PCIe Subsystem ID: As defined in PCIe 5.0 Base Specification~~
~~11-10~~		~~Reserved~~
~~19-12~~		~~Device Serial Number: Refer to definition of Device Serial Number Extended Capability in PCIe 5.0 Base Specification~~

~~2~~0	1	Ingress Port ID: Ingress management port index of the received request message. For CXL/PCIe ports, this corresponds to the physical port number. For non-CXL/PCIe, this corresponds to a vendor-specific index of the buses supported by the device, starting at 0. For example, a request received on the second of 2 SMBuses supported by a device would return a 1.
~~2~~1	1	Reserved
~~2~~2	1	Number of Physical Ports: Total number of physical ports in the CXL switch, including inactive/disabled ports
~~2~~3	1	Number of VCSs: Maximum number of virtual CXL switches supported by the CXL switch
~~55-24~~4	32	Active Port Bitmask: Bitmask defining whether a physical port is enabled (1) or disabled (0). Each bit corresponds 1:1 with a port, with the least significant bit corresponding to port 0
~~87-56~~36	32	Active VCS Bitmask: Bitmask defining whether a VCS is enabled (1) or disabled (0). Each bit corresponds 1:1 with a VCS ID, with the least significant bit corresponding to VCS 0
~~89-88~~68	2	Total Number of VPPBs: Maximum number of virtual PPBs supported by the CXL switch
~~91-90~~70	2	Number of Active VPPBs: Total number of VPPBs in use across all VCSs
~~92~~72	1	Number of HDM Decoders: Number of HDM decoders available per USP

Update section 8.2.9 as follows

Table 152. CXL Device Command Opcodes

Opcode					Required*	Input Payload Size (B)	Output Payload Size (B)
Command Set Bits[15:8]		Command Bits[7:0]		Combined Opcode			
00h	Information and Status	01h	Identify (Section 8.2.9.7.1)	0001h	PMB, MS	0	17
		02h	Background Operation Status (Section 8.2.9.7.2)	0002h	PMB, MS	0	8
		03h	Get Response Message Limit (Section 8.2.9.7.3)	0003h	PMB, MS	0	1
		04h	Set Response Message Limit (Section 8.2.9.7.4)	0004h	PMB, MS	1	1

…

*M = mandatory for all devices that implement ~~a Register Locator entry with Register Block Identifier =03h~~the CXL Device Register entry (Identifier=03h) in the Register Locator DVSEC (Section 8.1.9); PMB = prohibited on the primary and secondary mailboxes; MS = mandatory for components that support the FM API MCTP message type; O = Optional.

Add a new section, section 8.2.9.7 as follows

8.2.9.7 Information and Status Command Set

The Information and Status command set includes important commands for querying component capabilities and status.

8.2.9.7.1 Identify (Opcode 0001h)

Retrieves status information about the component, including whether it is ready to process commands. A component that is not ready to process commands shall return 'Retry Required'.

Possible Command Return Codes:

• Success

• Internal Error

• Retry Required

Command Effects:

• None

Table X Identify Output Payload

Byte Offset	Length	Description
1-0	2	PCIe Vendor ID: Identifies the manufacturer of the component, as defined in PCI Express Base Specification
3-2	2	PCIe Device ID: Identifier for this particular component assigned by the vendor, as defined in PCI Express Base Specification
5-4	2	PCIe Subsystem Vendor ID: Identifies the manufacturer of the subsystem, as defined in PCI Express Base Specification
7-6	2	PCIe Subsystem ID: Identifier for this particular subsystem assigned by the vendor, as defined in PCI Express Base Specification
15-8	8	Device Serial Number: Unique identifier for this device, as defined in the Device Serial Number Extended Capability in PCI Express Base Specification

16	1	Maximum Supported Message Size: The maximum supported size of the full message body (as defined in Figure 124) in bytes for any requests sent to this component, expressed as 2^n. The minimum supported size is 256 bytes (n=8) and the maximum supported size is 1 MB (n=20). This field is used by the caller to limit the Message Payload size such that the size of the Message Body does not exceed the capabilities of the component. The component shall discard any received messages that exceed the maximum size advertised in this field in a manner that prevents any internal receiver HW errors. The component shall return a response message with the 'Invalid Payload Length' return code for all received request messages that exceed the maximum size advertised in this field. The CXL specification guarantees that the size of the Identify Output Payload shall never exceed 244 Bytes (256 – 12 Bytes, the combined size of the fields preceding Message Payload).

8.2.9.7.2 Background Operation Status (Opcode 0002h)

Retrieve information about outstanding Background Operations processing on the interface from which this command was received.

Possible Command Return Codes:

• Success

• Internal Error

• Retry Required

• Invalid Payload Length

Command Effects:

• None

Table Y Background Operation Status Output Payload

Byte Offset	Length	Description
0	1	Background Operation Status: Reports the status of outstanding Background Operations: • Bit[0]: Background Operation – Indicates whether a background operation is in progress, as defined in Section 8.2.8.4.6. • Bits[7:1]: Percentage Complete – The percentage complete (0-100) of the background command, as defined in Section 8.2.8.4.7
1	1	Reserved
3-2	2	Command Opcode: The command identifier of the last command executed in the background. Refer to Section 8.2.9 for the list of command opcodes.
5-4	2	Return Code: The result of the command run in the background. Only valid when Percentage Complete = 100. Refer to Section 8.2.8.4.5.1.
7-6	2	Vendor Specific Extended Status: The vendor specific extended status of the last background command. Only valid when Percentage Complete = 100.

8.2.9.7.3 Get Response Message Limit (Opcode 0003h)

Retrieves the current configured response message limit used by the component. The component shall limit the output payload size of commands with variably sized outputs such that the full message body does not exceed the value programmed with this command.

Possible Command Return Codes:

- Success
- Internal Error
- Retry Required

Command Effects:

- None

Table Z Get Response Message Limit Output Payload

Byte Offset	Length	Description
0	1	Response Message Limit: The configured maximum size of the full message body for the response message generated by the component(as defined in Figure 124) in bytes, expressed as 2^n. The minimum supported size is 256 bytes (n=8) and the maximum supported size is 1 MB (n=20).

8.2.9.7.4 Set Response Message Limit (Opcode 0004h)

Configures the response message limit used by the component. The component shall limit the output payload size of commands with variably sized outputs such that the full message body does not exceed the value programmed with this command. Components shall return "Internal Error" if any errors within the component prevent it from limiting its response message size to a value lower than or equal to the requested size.

Possible Command Return Codes:

- Success
- Internal Error
- Retry Required
- Invalid Payload Length

Command Effects:

- None

Table A Set Response Message Limit Input Payload

Byte Offset	Length	Description
0	1	Response Message Limit: The configured maximum size of the full message body for response messages generated by the component (as defined in Figure 124) in bytes, expressed as 2^n. The minimum supported size is 256 bytes (n=8) and the maximum supported size is 1 MB (n=20).

Table B Set Response Message Limit Output Payload

Byte Offset	Length	Description
0	1	Response Message Limit: The configured maximum size of the full message body for response messages generated by the component (as defined in Figure 124) in bytes, expressed as 2^n. The value returned is the Response Message Limit used by the component after processing this request, which may be less than the requested value. The minimum supported size is 256 bytes (n=8) and the maximum supported size is 1 MB (n=20). The FM shall discard any messages sent by the component that exceed the maximum size set with this field in a manner that prevents any internal receiver HW errors.

F25 Support for Components with Limited Buffering Capability

Update section 7.6.7.2.1 as follows

This command retrieves information on a specified number of VCSs in the switch. Due to the possibility of variable numbers of vPPBs in each VCS, the returned array has variably sized elements.

Possible Command Return Codes:

- Success
- Invalid Parameter
- Internal Error
- Retry Required
- Invalid Payload Length

Command Effects:

- None

Table 97. Get Virtual CXL Switch Info Request Payload

Bytes	Description
0	Start vPPB: Specifies the ID of the first vPPB requested in the vPPB information list (bytes 4 – 7 in Table 99). This enables compatibility with devices that have small maximum command message sizes.

1	vPPB List Limit: The maximum number of vPPB information entries to include in the response (bytes 4 – 7 in Table 99). This enables compatibility with devices that have small maximum command message sizes.
~~0~~2	Number of VCSs: Number of VCSs requested.
~~Varies 1~~(Number of VCSs + 2) - 3	VCS ID List: 1 byte ID of requested VCS, repeated Number of VCSs times.

…

Table 99. Get Virtual CXL Switch Info VCS Information Block Format

Bytes	Description
0	Virtual CXL Switch ID
1	VCS State: Current state of the VCS: 0 – Disabled 1 – Enabled 2 to 0xFE – Reserved 0xFF – Invalid VCS_ID; all subsequent field values are invalid
2	USP ID: Physical port ID of the CXL switch for the Upstream Port
3	Total Number of vPPBs: Total number of vPPBs in the VCS. This value may be larger than the 'vPPB List Limit' field specified in the request. In this case, the length of vPPB information list, starting at byte 4, is defined by 'vPPB List Limit', not this field.
4	PPB[~~0~~Start vPPB] Binding Status: 0 – Unbound 1 – Bind or unbind in progress 2 – Bound Physical Port 3 – Bound LD
5	PPB[~~0~~Start vPPB] Bound Port ID: Physical port number of bound port.
6	PPB[~~0~~Start vPPB] Bound LD ID: ID of LD bound to port from MLD on associated physical port. Only valid if VPPB[~~0~~Start vPPB]_Status is 3, 0xFF otherwise.
7	Reserved
…	
4 + (Number of vPPBs* - 1) * ~~3~~4	PPB[Start vPPB + Number of vPPBs* - 1] Binding Status: 0 – Unbound

	1 – Bind or unbind in progress 2 – Bound Physical Port 3 – Bound LD
5 + (Number of vPPBs* - 1) * ~~3~~4	PPB[Start vPPB + Number of vPPBs* - 1] Bound Port ID: Physical port number of bound port.
6 + (Number of vPPBs* - 1) * ~~3~~4	PPB[Start vPPB + Number of vPPBs* - 1] Bound LD ID: ID of LD bound to port from MLD on associated physical port. Only valid if PPB[Start vPPB + Number of vPPBs* - 1] Binding Status is "Bound LD", 0xFF otherwise.
7 + (Number of vPPBs* - 1) * ~~3~~4	Reserved

* The vPPB information list length is defined by the lesser of the 'vPPB List Limit' field in the command request and the 'Number of vPPBs' field in the command response

…

Update section 7.6.7.5.2 as follows

This command gets the memory allocations of the MLD.

Possible Command Return Codes:

- Success
- Invalid Parameter
- Unsupported
- Internal Error
- Retry Required
- Invalid Payload Length

Command Effects:

- None

Table Z Get LD Allocations Request Payload

Bytes	Description
0	Start LD ID: ID of the first LD in the LD Allocation List
1	LD Allocation List Limit: Maximum number of LD information blocks returned

Table 112. Get LD Allocations Response Payload

Bytes	Description
0	Number of LDs: Number of LDs enabled in the device ~~information blocks returned~~.
1	Memory Granularity: - This field specifies the granularity of the memory sizes configured for each LD:

	0h - 256 MB 1h - 512 MB 2h - 1 GB All others - Reserved
2	Start LD ID: ID of the first LD in the LD Allocation List
3-~~2~~	~~Reserved~~LD Allocation List Length: Number of LD information blocks returned. This value is the lesser of the request's 'LD Allocation List Limit' and responses 'Number of LDs'.
Varies-4	LD Allocation List: LD Allocation blocks for each LD, as defined in Table 113, repeated LD Allocation List Length~~Number of LDs~~ times.

…

Update section 7.6.7.5.3 as follows

Possible Command Return Codes:

- Success
- Invalid Parameter
- Unsupported
- Internal Error
- Retry Required
- Invalid Payload Length

Command Effects:

- None

Table 114 Set LD Allocations Request Payload

Bytes	Description
0	Number of LDs: Number of LDs to configure
1	Start LD ID: ID of the first LD in the LD Allocation List
3-~~1~~2	Reserved
Varies-4	LD Allocation List: LD Allocation blocks for each LD starting at Start LD ID, as defined in Table 113, repeated Number of LDs times.

Table 115. Set LD Allocations Response Payload

Bytes	Description
0	Number of LDs: Number of LDs configured
1	Start LD ID: ID of the first LD in the LD Allocation List

3-~~1~~2	Reserved
Varies-4	LD Allocation List: Updated LD Allocation blocks for each LD starting at Start LD ID, as defined in Table 113, repeated Number of LDs times.

Update section 7.6.7.5.7 as follows

Table A Payload for Get QoS Allocated BW Request

Bytes	Description
0	Number of LDs: Number of LDs configured
1	Start LD ID: ID of the first LD in the QoS Allocated BW List

Table 118. Payload for Get QoS Allocated BW Response, Set QoS Allocated BW Request, and Set QoS Allocated BW Response

Bytes	Description
0	Number of LDs: Number of LDs configured
1	Start LD ID: ID of the first LD in the QoS Allocated BW List
~~(n-1) 0~~(Number of LDs + 2) - 2	QoS Allocation Fraction: Byte array of allocated bandwidth fractions for LDs starting at Start LD ID~~, where n = LD Count, as returned by the Get LD Info command~~. The valid range of each array element is 0-255. Default value is 0. Value in each byte is the fraction multiplied by 256.

Table B Payload for Get QoS BW Limit Request

Bytes	Description
0	Number of LDs: Number of LDs configured
1	Start LD ID: ID of the first LD in the QoS BW Limit List

Table 119. Payload for Get QoS BW Limit Response, Set QoS BW Limit Request, and Set QoS BW Limit Response

Bytes	Description
0	Number of LDs: Number of LDs configured
1	Start LD ID: ID of the first LD in the QoS BW Limit List
~~(n-1)-0~~(Number of LDs + 2) - 2	QoS Limit Fraction: Byte array of allocated bandwidth limit fractions for LDs starting at Start LD ID~~, where n = LD Count, as returned by the Get QoS BW command~~. The valid range of each array element is 0-255. Default value is 0. Value in each byte is the fraction multiplied by 256.

Update section 8.2.8.4.5.1 as follows

Table 150. Command Return Codes

Value	Definition
…	
0016h	Invalid Payload Length: The input payload length specified i~~n the Command Register~~for the command is not valid or exceeds the component's Maximum Supported Message Size. The device is required to perform this check prior to processing any command defined in this Specification.

Update section 8.2.9.4.1 as follows

8.2.9.4.1 Get Supported Logs (Opcode 0400h)

Table C Get Supported Logs Input Payload

Byte Offset	Length	Description
0	1	Maximum Number of Supported Log Entries: Maximum number of Supported Log Entries requested
1	1	Start Log Entry Index: Index of the first Supported Log Entry requested

Table 168. Get Supported Logs Output Payload

Byte Offset	Length	Description
0	2	Number of Supported Log Entries: The number of Supported Log Entries returned in the output payload

2	2	Total Number of Supported Log Entries: The total number of Supported Log Entries supported by the component
4	1	Start Log Entry Index: Index of the first Supported Log Entry in the output payload
~~2~~5	~~6~~3	Reserved
8	Varies	Supported Log Entries: Device specific list of supported log identifier UUIDs and the current size of each log.

F26 Incorrect Table Links

Update section 7.6.7.5.8 as follows

Payload for Set QoS Allocated BW Request and Response is documented in Table 118~~6~~.

Update section 7.6.7.5.10 as follows

Payload for Set QoS BW Limit Request and Response is documented in Table 119~~6~~.

F27 FM API Event Notifications

Update section 7.6.2 as follows

~~The FM API consists of request messages, response messages and event notification messages.~~ FMs issue request messages and CXL components issue response ~~and event notification~~ messages. CXL components may also issue the "Event Notification" request if notifications are supported by the component and the FM has requested notifications from the component using the Set OOB Event Interrupt Policy command. Refer to Section 7.6.3 for transport protocol details. MCTP may be used as the transport protocol.

…

Update section 7.6.3 as follows

Table 84. FM API Message Format

Bytes	Description
0	Bits (3:0): Message Category: Type of FM API message: • 0h = Request

	• 1h = Response ~~• 2h = Event Notification~~ • All other encodings reserved Bits (7:4): Reserved
1	Message Tag: Tag number assigned to request messages by requesters. The message tag shall be initialized to 0 and increment by 1 for each new request, rolling over to 0 after it reaches FFh. The tag number may be used to track response messages and to identify the retransmission of Event Notification requests. Response messages shall use the tag number from the corresponding Request message. ~~Must be 0 for Event Notification messages.~~
2	Reserved
4-3	Command Opcode: As defined in Table 152 and Table 205.
7-5	Bits (20:0): Message Payload Length - As defined in Table 152 and Table 205. Bit (22:21): Reserved Bit(23): Background Operation: As defined in Section 8.2.8.4.6. ~~Must be 0 for Request messages and Event Notification Messages.~~
9-8	Return Code: As defined in Table 150. Must be 0 for Request messages and Event Notification Messages
11-10	Vendor Specific Extended Status: As defined in Section 8.2.8.4.6. ~~Must be 0 for Request messages and Event Notification Messages~~
Varies-12	Message Payload: The length of this field is specified in the Message Payload Length field above. The format depends on Opcode and Message Category, as defined in Section 7.6.7

Update section 7.6.6.4 as follows

1. To facilitate some system operations, the FM requires event notifications so it can execute its role in the process in a timely manner (e.g., notifying hosts of an asserted Attention Button on an MLD during Managed Hot-Removal). If supported by the device, the FM can check the current event notification settings with the Get OOB Event Interrupt Policy command and modify them with the Set OOB Event Interrupt Policy command.
2. If supported by the component, the event logs can be read with the Get Event Records command to check for any error events experienced by the component that might impact normal operation.

Update section 7.6.7.1 as follows

7.6.7.1 Switch Event Notifications ~~Command Set~~

Switches use the Event Notification command, as defined in 8.2.9.1.8, to issue interrupts to the FM.

This optional command set is used by devices to send notifications to the FM. The following commands are defined:

Table 86. Switch Event Notifications Command Set Requirements

Command Name	Requirement
Event Notification	O

*O = Optional

7.6.7.1.1 Event Notification (Opcode 5000h)

This command is used by a CXL device to send notifications to the FM. It is only sent by CXL devices. Any commands of this type received by CXL devices should be silently discarded. There is no response for this command, it is a notification to the FM that there are either new events to be read from the Event Records or that the FM must initiate other management activities. The FM acknowledges a notification by clearing it with the Manage Events command returning a response.

A single notification using the same Message Tag is sent every 10 ms after the last notification was sent until the FM has cleared all event records.

Possible Command Return Codes:

- Success
- Invalid Parameter
- Internal Error
- Retry Required

Command Effects:

- None

Table 87. Event Notification Payload

Bytes	Description
0	Event Log: The specific device event logs generating the notification 00h = Informational Event Log 01h = Warning Event Log 02h = Failure Event Log 03h = Fatal Event Log Other values reserved.
7:1	Reserved
131:8	Event Record

Update section 8.2.9 as follows

Table 152. CXL Device Command Opcodes

Opcode					Required*	Input Payload Size (B)	Output Payload Size (B)
Command Set Bits[15:8]		Command Bits[7:0]		Combined Opcode			
…							
01h	Events	00h	Get Event Records (Section 8.2.9.1.2)	0100h	M	1	20h+
		01h	Clear Event Records (Section 8.2.9.1.3)	0101h	M	8+	0
		02h	Get Event Interrupt Policy (Section 8.2.9.1.4)	0102h	M	0	4
		03h	Set Event Interrupt Policy (Section 8.2.9.1.5)	0103h	M	4	0
		04h	Get OOB Event Interrupt Policy (Section 8.2.9.1.6)	0104h	PMB, O	2	0
		05h	Set OOB Event Interrupt Policy (Section 8.2.9.1.7)	0105h	PMB, O	2	2
		06h	Event Notification (Section 7.6.7.1.8)	0106h	PMB, O	2	0
…							

*M = mandatory for all devices that implement ~~a Register Locator entry with Register Block Identifier = 03h~~the CXL Device Register entry (Identifier=03h) in the Register Locator DVSEC (Section 8.1.9); PMB = prohibited on the primary and secondary mailboxes; O = Optional.

Add sections 8.2.9.1.6, 8.2.9.1.7 and 8.2.9.1.8 as follows

8.2.9.1.6 Get OOB Event Interrupt Policy (Opcode 0104h)

Retrieve the settings for interrupts that are signaled for component events over OOB management media. When notifications are enabled for a particular log on a component, the component may issue up to 1 Event Notification for that log type when the log contents transition from zero entries to one or more entries. The FM must be able to receive at least 1 Event Notification message from the component for each log for which interrupts have been enabled.

Possible Command Return Codes:

- Success
- Internal Error
- Retry Required
- Invalid Payload Length

Command Effects:

- None

Table XX Payload for Get OOB Event Interrupt Policy Output, Set OOB Event Interrupt Policy Input, and Set OOB Event Interrupt Policy Output

Byte Offset	Length	Description
0	2	Event Interrupt Settings: Bitmask indicating whether event notifications are enabled (1) or disabled (0) for a particular event Bit[0]: New uncleared Informational Event Log record(s) Bit[1]: New uncleared Warning Event Log record(s) Bit[2]: New uncleared Failure Event Log record(s) Bit[3]: New uncleared Fatal Event Log record(s) Bits[15-4]: Reserved

8.2.9.1.7 Set OOB Event Interrupt Policy (Opcode 0105h)

Change the settings for the interrupts that are signaled for component events. The receiver may capture the address of the requesting component in a transport specific way. The subsequent enabled events shall be sent to that address.

Interrupts shall only be generated for events that occur after this command is received by the component with input parameters that enable logging for the corresponding log type. Components should immediately terminate the retransmission of Event Notification requests that have not been acknowledged if a request of this type has been received with input parameters that disable logging for the corresponding log type.

Possible Command Return Codes:

- Success
- Invalid Parameter
- Internal Error
- Retry Required
- Invalid Payload Length

Command Effects:

- Immediate Policy Change

Input and output payloads for this command are documented in Table XX.

8.2.9.1.8 Event Notification (Opcode 0106h)

This command is used by a CXL component to send notifications to the FM. It is only sent by CXL components. Any commands of this type received by CXL components shall be silently discarded.

This command is a notification to the FM that there are new events to be read from the event logs specified in the "Event Log" payload field. The notification is triggered when both of the following conditions are met: the log transitions from having no entries to having one or more entries and OOB interrupts have been enabled with the Set OOB Event Interrupt Policy command. Log entries are cleared with the Clear Event Logs command. After clearing log entries, the FM should use the Get Event Records command to confirm that all entries have been cleared from the log to ensure that the notification trigger has been re-armed.

The FM acknowledges a notification by returning a response. The component shall retransmit 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.

Possible Command Return Codes:

- Success
- Invalid Parameter
- Internal Error
- Retry Required

Command Effects:

- None

Table Y Event Notification Input Payload

Byte Offset	Length	Description
0	2	Event: Bitmask indicating the event: Bit 0 = Informational Event Log has uncleared record(s) Bit 1 = Warning Event Log has uncleared record(s) Bit 2 = Failure Event Log has uncleared record(s) Bit 3 = Fatal Event Log has uncleared record(s) Bits 15-4 = Reserved

Update sections 8.2.9.6 as follows

Table 205. CXL FM API Command Opcodes

Opcode	Required*

Command Set Bits[15:8]		Command Bits[7:0]		Combined Opcode	
~~50h~~	~~Switch Event Notifications~~	~~00h~~	~~Event Notification (Section 7.6.7.1.1)~~	~~5000h~~	~~M~~
…					

*MSW = mandatory for all switches that support the FM API MCTP message type, PSW = Prohibited for Switches, OSW = Optional for Switches, MMLD = Mandatory for all MLD components, PMLD = Prohibited for all MLD components, OMLD=Optional for all MLD components.

F28 CEDT CFMWS & QTG DSM ECN Errata

Update section 9.14.1.3 in CEDT CFMWS & QTG DSM *as follows*

..	..	..	..
Interleave Target List	4 * NIW ~~HBIW~~	36	.. The set of HPAs decoded by Entry N in the Interleave Target List shall satisfy the following equations 1. Base HPA <= HPA < Base HPA + Windows Size 2. If (Interleave Arithmetic==0) ~~Floor[(HPA – Base HPA)/HBIG] MOD NIW = N; where "/" represents a Division operation. MOD is a standard Modulo operation and Floor function returns the largest Integer that is not greater than the input.~~ N = HPA[8+HBIG+NIW-1:8+HBIG] N is 0 based (0<= N <NIW).

F29 Appendix B, Type-2 Memory Request Table

In Appendix B, Table 311, there are two rows that should have indicated "I" state as next Bias State instead of UC. This errata is just showing the section of the table where this correction is needed. If an implementation follows the "UC" (Un-Changed) encoding the result may be bias state that is conservative by leaving bias state as S or A resulting in an unnecessary bias flip from the device.

Table 311. Type2 Memory Request

Legal	Host Request				Device Response				Final Device State		Description
	M2S Req	Meta Field	Meta Value	Snp Type	S2M NDR	S2M DRS	Meta Field	Meta Value	Device Cache	Bias State	
Y(1)	MemRd	MS0	A	SnpInv	Cmp-E	MemData	<any>	<any>	I	A	The Host wants an exclusive copy of the line
N				SnpData							
N				SnpCur							
N				No-Op							
N			S	SnpInv							
Y(1)				SnpData	Cmp-S		<any>	<any>	S	S	The Host requesting a shared copy of the line, but Rsp types allow device to return S or E state to host. Cmp-E response is not recommended because device did not request this state.
Y(1)					Cmp-E		<any>	<any>	I	A	
N				SnpCur							
N				No-Op							
Y			I	SnpInv	Cmp		<any>	<any>	I	~~UC~~ I	The Host requesting a non-cacheable but current value of the line and forcing device to flush its cache.
				SnpData							
Y				SnpCur	Cmp		<any>	<any>	<any>	~~UC~~ I	The Host requesting a non-cacheable but current value of the line leaving data in the device's cache.
N				No-Op							
Y(1)		No-Op	NA	SnpInv	Cmp		<any>	<any>	I	UC	The Host wants to read line without changing state expected in the host cache and the device should invalidate the line from its cache.
				SnpData							
Y(1)				SnpCur	Cmp		<any>	<any>	<any>	UC	The Host wants a current value of the line without changing the state expected in the host cache.
Y				No-Op	Cmp		<any>	<any>	<any>	UC	Host wants a the value of the memory location without snooping the device cache and without changing cache state expected in the host cache. A use case for this would be if the host includes E or S-state without data so it is requesting data only and doesn't want to change cache state and because it has E or S state it can know that the device cache does not need to be snooped.

F30 Incorrect passing criteria in CXL Capability Header test

In section 14.13.1, please make the following change to the pass critera:

Pass Criteria:

- Test ~~15.6.4~~ 14.8.2 Passed.
- Verify Conditions Met

F31 Update Reference to CDAT Specification

Add new rows to Table 2. Reference Documents

Document	Chapter Reference	Document No./Location
..		
Coherent Device Attribute Table (CDAT) Specification, version 1.02 or later	Chapters 8, 9, 14	https://www.uefi.org/acpi

Delete the footnote under section 8.1.11 as follows

8.1.11 Table Access DOE

Coherent Device Attributes Table~~1~~ (CDAT) allows a device or switch to expose its performance attributes such as latency and bandwidth ..

~~1. See https://www.uefi.org/uefi and ACPI Specification~~

F32 Eliminate the term "Host Space"

In Table 5, make the following change:

Field	Description	Note
..		
Parameter[15:0]	.. RESETPREP (Request and Response): [7:0] - ResetType 0x01 => ~~host space~~ System transition from S0 to S1; 0x03 => ~~host space~~ System transition from S0 to S3; 0x04 => ~~host space~~ System transition from S0 to S4; 0x05 => ~~host space~~ System transition from S0 to S5; 0x10 => ~~Host space~~ System reset ~~(host space partition reset)~~ ..	..

In section 9.3, make the following change:

..

During system reset flow, host shall issue a CXL PM VDM (see Table 5) to the downstream CXL components with the following values.

- ..
- ResetType = ~~Warm Reset~~ System reset
- ..

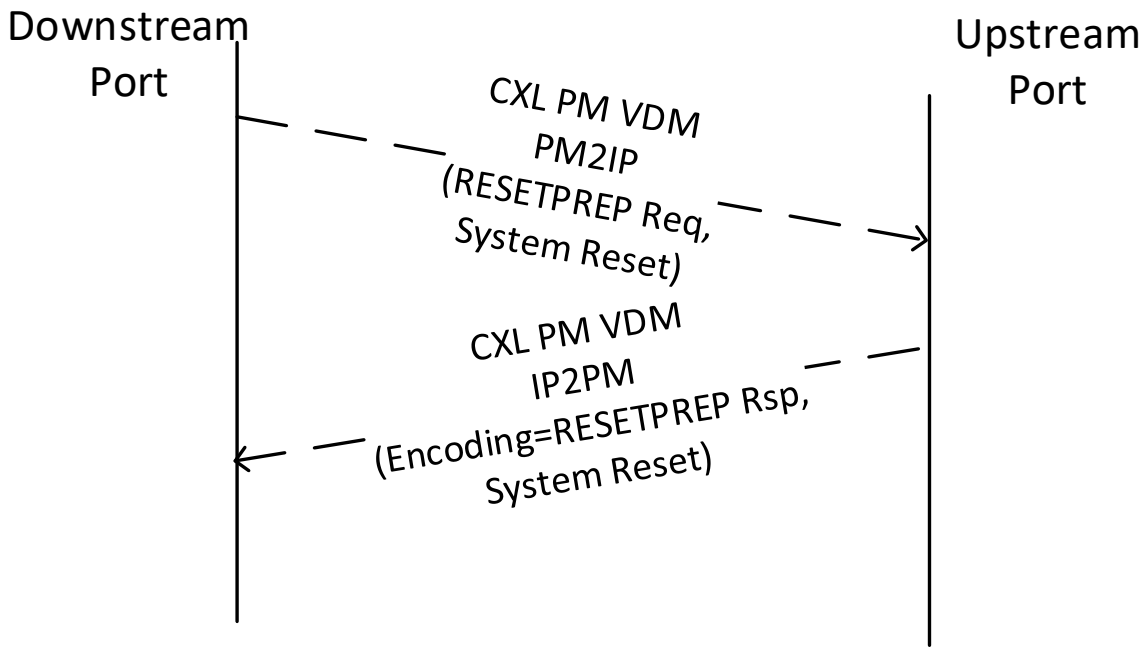
~~..~~

.. After all the Reset preparation is completed, the CXL device shall issue a CXL PM VDM with the following value

- ..
- ResetType = ~~Warm Reset~~ System reset
- ..

..

Replace Figure 141 to



In section 9.4, make the following change:

..

During Sx flow, host shall issue a CXL PM VDM (see Table 5) to the downstream CXL components with the following values.

- ..
- ResetType = ~~host space~~ System transition from S0 to Sx (S1, S3, S4 or S5)
- ..

..

.. After all the Reset preparation is completed, the CXL device shall issue a CXL PM VDM with the following value

- ..
- ResetType = ~~host space~~ System transition from S0 to Sx (S1, S3, S4 or S5)
- ..

..

F33 Chapter 11, Figure 177 and Figure 179

As part of the Truncated MAC (TMAC) flows discussed CXL2.0 Section 11.1.9 there are two figures which show examples of the flits sent as part of the TMAC flow. The figures have created confusion because they show "IDLE" as the only flits following the sending of TMAC, and before the TMAC it shows "IDLE or CONTROL". To avoid deadlock following the TMAC flit, either IDLE or other control flits are allowed (no protocol flits). This errata will correct the figures where red "x" shows the old text which is replace with text to the right of the "x".

Figure 177. Early Termination and Transmission of Truncated MAC Flit

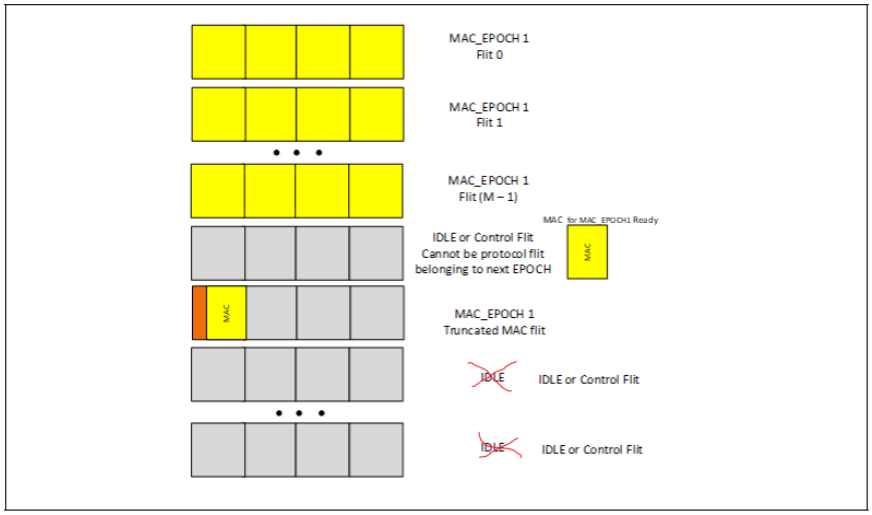


Figure 179. Link Idle Case After Transmission of Aggregation Flit Count Number of Flits