Compute Express Link™ (CXL™)

Engineering Change Notice to the Specification 2.0

July 2021

Compliance ECN – Memory Device Error Injection

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).

CXL ENGINEERING CHANGE NOTICE

TITLE:	Compliance ECN – Memory Device Error Injection
DATE:	Introduced date (12/03/2020) Updated date (03/10/2021)
AFFECTED DOCUMENT:	CXL 2.0 – Chapter 14
SPONSOR:	Chet Douglas – Intel Corporation

Part I

1. **Summary of Functional Changes**
   Provide additional optional compliance DOE interfaces to inject a class of errors that are specific to CXL.mem devices. This includes injecting poison errors into the device's host-addressable media or Label Storage Area (LSA) and injecting Memory Device Health status changes.

2. **Benefits as a Result of the Changes**
   The CXL compliance chapter was missing critical injection capabilities for CXL.mem devices. Without this ECN, vendors would be forced to design vendor specific mechanism to inject errors, defeating the purpose of a common compliance interface.

3. **Assessment of the Impact**
   Device's will have the option to implement additional DOE interfaces to provide the new functionality proposed here.

4. **Analysis of the Hardware Implications**
   Device's will have the option to implement additional DOE interfaces to provide the new functionality proposed here.

5. **Analysis of the Software Implications**
   Compliance Software will need to implement the additional test scripts defined in this ECN to test the device's error injection and error handling capabilities through the compliance interface.

6. **Analysis of the Compliance and Test Implications**
   This section describes new changes required to Compliance and Test specifications to cover this feature. Other than the proposed subsection name changes, there should be no impact to existing compliance tests.

Part II

Detailed Description of the change

Add/modify the following under CXL Spec 2.0

The following changes are in the compliance section 14

Updated section 14.16.4.2 with additional status values

14.16.4.2 Compliance Mode Status

Table 301. Compliance mode Return Values

Value	Description
0x00000000	Success
0x00000001	Not Authorized
0x00000002	Unknown Failure
0x00000003	Unsupported injection function
0x00000004	Internal Error
0x00000005	Target busy
0x00000006	Target Not Initialized
0x00000007	*Invalid Address Specified*
0x00000008	*Invalid Injection Parameter*

Rename section 14.16.4.7 from "Inject Poison" to "Inject Link Poison"

New section added to 14.16.4.17:

14.16.4.17 Inject Memory Device Poison

Table XXX. Enable Memory Device Media Poison Injection into

Data Object Byte Offset	Length	Description
0h	8	Standard DOE Request Header
8h	1	Request Code = 10h, Memory Device Media Poison Injection
9h	1	Version
0Ah	2	Reserved
0Ch	1	Protocol 2= mem
0Dh	1	Reserved
0Eh	1	Action 0=Inject Poison, 1=Clear Poison
0Fh	1	Reserved
10h	8	Device Physical Address – When Protocol = 2, the device shall inject poison into the media at this requested address. If this address specifies a persistent memory address, the injected poison shall persist across warm or cold resets. Device shall report Invalid Address Specified poison injection response status if the DPA is out of range. • Bits[5:0]: Reserved • Bits[7:6]: DPA[7:6] • Bits[15:8]: DPA[15:8] • … • Bits[63:56]: DPA[63:56]
18h	8	Clear Poison Write Data – When Protocol = 2 and Action = 1, the device shall write this replacement data into the requested physical address, atomically, while clearing poison.

Table XXX. Enable Memory Device LSA Poison Injection into

Data Object Byte Offset	Length	Description
0h	8	Standard DOE Request Header
8h	1	Request Code = 11h, Memory Device LSA Poison Injection
9h	1	Version
0Ah	2	Reserved
0Ch	1	Protocol 2= mem
0Dh	1	Reserved
0Eh	1	Action 0=Inject Poison, 1=Clear Poison
0Fh	1	Reserved
10h	4	LSA Byte Offset – When Protocol = 2, the device shall inject poison into the Label Storage Area of the device at this requested byte offset. Since the LSA is persistent, the injected poison shall persist across warm or cold resets. Device shall report Invalid Address Specified poison injection response status if the byte offset is out of range. Clearing the poison can be done through this interface or with SetLSA.

New section added 14.16.4.18:

14.16.4.18 Inject Memory Device Health

Table XXX. Enable Memory Device Health Injection

Data Object Byte Offset	Length	Description
0h	8	Standard DOE Request Header
8h	1	Request Code = 11h, Memory Device Health Injection
9h	1	Version
0Ah	2	Reserved
0Ch	1	Protocol 2= mem
0Dh	1	Injection Type 0=Error is injected immediately and remains in effect until it is cleared using this command or by a CXL warm or cold reset of the device, 1 = Error is not injected until after a cold reset, the injection will only occur once, and be auto-disabled after the first occurrence
0Eh	1	Valid Device Health Injection: Indicators of what Device Health Injection fields are valid in the supplied in the payload • Bit[0]: When set, the Health Status Injection Enabled field shall be valid. • Bit[1]: When set, the Media Status Injection Enabled field shall be valid. • Bit[2]: When set, the Life Used Injection Enabled field shall be valid • Bit[3]: When set, the Dirty Shutdown Count Injection Enabled field shall be valid • Bit[4]: When set, the Device Temperature Injection Enabled field shall be valid • Bits[7:5]: Reserved
0Fh	1	Enable Device Health Injection: The device shall enable the following error injection. • Bit[0]: Health Status Injection Enabled -When set, the Health Status field shall be valid and the device shall enable its Health Status injection. When clear, the device shall disable its Health Status injection. • Bit[1]: Media Status Injection Enabled - When set, the Media Status field shall be valid and the device shall enable its Media Status injection. When clear, the device shall disable its Media Status injection. • Bit[2]: Life Used Injection Enabled - When set, the Life Used field shall be valid and the device shall enable its Life Used injection. When clear, the device shall disable its Life Used injection. • Bit[3]: Dirty Shutdown Count Injection Enabled - When set, the Dirty Shutdown Count field shall be valid and the device shall enable its Dirty Shutdown Count injection. When clear, the device shall disable its Dirty Shutdown Count injection. • Bit[4]: Device Temperature Injection Enabled - When set, the Device Temperature field shall be valid and the device shall enable its Device Temperature injection. When clear, the device shall disable its Device Temperature injection. Bits[7:5]: Reserved
10h	1	Health Status – The injected Health Status. One of the defined Get Health Info values from the Memory Device Commands section. Return Invalid Injection Parameter for invalid or unsupported injection values.
11h	1	Media Status – The injected Media Status. One of the defined Get Health Info values from the Memory Device Commands section. Return Invalid Injection Parameter for invalid or unsupported injection values.
12h	1	Life Used – The injected Life Used. See the Get Health Info command in the Memory Device Commands section for legal range. Return Invalid Injection Parameter for invalid or unsupported injection values.
13h	1	Reserved
14h	4	Dirty Shutdown Count – The injected Dirty Shutdown Count. See the Get Health Info command in the Memory Device Commands section. Return Invalid Injection Parameter for invalid or unsupported injection values.
18h	2	Device Temperature – The injected Device Temperature. See the Get Health Info command in the Memory Device Commands section. Return Invalid Injection Parameter for invalid or unsupported injection values.

Table XXX. Device Health Injection Response

Data Object Byte Offset	Length	Description
0h	8	Standard DOE Request Header
8h	1	Response Code = 10, Device Health Injection
9h	1	Version of Capability Returned
0Ah	1	Length of Capability Package
0Bh	1	Status: See table 301 for error codes

Rename 14.12.1.5 from “CXL.mem Poison Injection” to “CXL.mem Link Poison Injection”

New section 14.12.1.10 CXL.mem Media Poison Injection

14.12.1.10 CXL.mem Media Poison Injection

14.12.1.10.XX Host to Memory Device Poison Injection

Test Steps:

1. Select error injection target address Device Physical Address (DPA) that belongs to the Device Under Test.
2. Translate DPA to Host Physical Address (HPA)
3. Request Poison error injection via Enable Memory Device Poison Injection DOE specifying the DPA where the error is to be injected.
4. Poll on the Poison Injection Response DOE. Successful completion status indicates the device has injected the poison into the memory.
5. Host performs a memory read at the error injection target HPA and the device responds to the read with the poison indicator set.

Pass Criteria:

- Receiver (Device) logs poisoned received error.
- When injecting poison into persistent memory regions of the CXL.mem device:
    - The device shall add the new physical address to the device’s poison list and the error source should be set to an injected error and reported through the Get Poison List command.
    - In addition, the device should add an appropriate poison creation event to its internal Informational Event Log, update the Event Status Register, and if configured interrupt the host.
    - Poison shall be persistent across warm or cold reset until explicitly cleared by overwriting the cache line with new data with the poison indicator cleared.

Fail Criteria:

- Receiver does not log poison received error

New section 14.12.1.11 CXL.mem LSA Poison Injection

14.12.1.11.XX Host to Memory Device LSA Poison Injection

Test Steps:

1. Select error injection LSA byte offset that belongs to the Device Under Test.

2. Request LSA Poison error injection via Enable Memory Device LSA Poison Injection Compliance DOE specifying the LSA byte offset where the error is to be injected.

4. Poll on the Poison Injection Response DOE. Successful completion status indicates the device has injected the poison into the memory.

5. Host performs a GetLSA mailbox command including the LSA byte offset where the poison was injected in to the LSA and the device responds to the read with an error in the mailbox GetLSA command and appropriate error log generation.

Pass Criteria:

- Receiver (Device) errors the GetLSA command to the injected LSA byte offset.
- When injecting poison into the persistent memory Label Storage Area of the CXL.mem device:
  - The device should add an appropriate poison creation event to its internal Informational Event Log, update the Event Status Register, and if configured interrupt the host.
  - Poison shall be persistent across warm or cold reset until explicitly cleared by a SetLSA with new data that over-writes the poisoned data at the original poison injection LSA byte offset.

Fail Criteria:

• Receiver does not log poison received error

New section 14.12.1.12 CXL.mem Health Injection

14.12.1.12 CXL.memory Device Health Injection

This test is only applicable if a device supports the Device Health Injection with the DOE transport.

14.12.1.X.1 Host to Device Poison Injection

Test Steps:

1. Request device health injection via Enable CXL.memory Device Health Injection

Compliance DOE specifying the health status field to inject.

2. Poll on the Poison Injection Response DOE. Successful completion status indicates the device has injected the health status change into the device.
3. Host verifies device health status changes by inspecting event log event records and device health status changes.

Pass Criteria:

- Device notifies host of state change through appropriate Event Log Event Records and the resulting change in device health can be verified through Get Health Info command.

Fail Criteria:

• Receiver does not see correct event logs or change in health status.