CXL ENGINEERING CHANGE NOTICE

| | |
|---|---|
| TITLE: | Compatibility with the PCIe MMPT ECN |
| DATE: | Introduced date (12/12/2023)<br>Updated date (3/18/2024)<br>• Fixed typo in section 8.2.8.4.3 Mailbox Capabilities Register in the Type field description.<br>Updated date (3/7/2024)<br>• In sections 8.2.8.1 CXL Device Capabilities Array Register and 8.2.8.4.3 Mailbox Capabilities Register, add clarification in the Type field that CXL only defines bits 26:24.<br>Updated date (2/2/2024)<br>• Add PCIe Base Specification v6.2 as [PCIe] and changed references from [PCIe MMPT ECN] to [PCIe].<br>• Updated Implementation note in Section 8.2.8 to clarify how updated SW enumeration of CXL defined capabilities.<br>• Changed field definitions from "Reserved" to "Refer to [PCIe] for the definition of this field" for clarity.<br>• Restored statement that mailbox can be reset via CXL reset.<br>Updated date (1/9/2024)<br>• Fixed the PCIe MMB MCAP ID to 01h in the implementation note in Section 9.20. |
| AFFECTED DOCUMENT: | CXL v3.1 |
| SPONSOR: | Intel, Dell, Lenovo |

## Part I

[PCI MMPT ECN] = https://members.pcisig.com/wg/PCI-SIG/document/20109
[PCIe] = PCIe Base Specification v6.2 (link TBD)

### 1. Summary of Functional Changes

The [PCIe MMPT ECN] introduced in-band management interfaces that can be used by system software to manage PCIe Functions, including CXL components. These interfaces were based off of and are backwards compatible with existing CXL in-band management interfaces as described in the following table.

| **PCIe MMPT ECN Interface** | **CXL Interface** |
|---|---|
| MMIO Register Block Locator (MRBL)<br>([PCI MMPT ECN] Section 7.9.30) | Register Locator DVSEC<br>(Section 8.1.9) |
| MMIO Capabilities (MCAP) Register Block<br>([PCI MMPT ECN] Section 6.35.1) | CXL Device Register Interface<br>(Section 8.2.8) |
| MMIO Mailbox (MMB) Capability<br>([PCI MMPT ECN] Section 6.35.1.3) | Mailbox Registers<br>(Section 8.2.8.4) |
| MMB Command Interface<br>([PCI MMPT ECN] Section 6.36) | Component Command Interface<br>(Section 8.2.9) |

This ECN enables CXL to transition to the PCIe defined interfaces for new functionality while maintaining compatibility with existing CXL implementations. It includes the following changes:

1. Add guidance in Section 8.2.8 for CXL components and software on supporting the [PCIe MMPT ECN] while maintaining compatibility with existing CXL implementations.
2. Update the Register definitions in Section 8.2.8 to ensure compatibility with the PCIe defined registers.
3. Update the Component Command Interface Section 8.2.9 to provide implementation guidance on supporting the [PCIe MMPT ECN] MMB Command Interface.
4. Update the Component Command Interface Section 9.20 to add an implementation note regarding compatibility between the CXL mailbox and the PCIe MMB.

**2. Benefits as a Result of the Changes**

Supporting the [PCIe MMPT ECN] allows CXL components to adopt new management capabilities that are generic to all types of PCIe Functions. For example, the MMPT command set supports tunneling commands defined in other industry standards (e.g., DMTF PLDM) over the mailbox interface.

**3. Assessment of the Impact**

Supporting the [PCIe MMPT ECN] is optional and therefore implementing the changes described in this ECN is optional for existing CXL implementations. However, it is expected that new functionality will utilize the [PCIe MMTP ECN] interfaces.

**4. Analysis of the Hardware Implications**

To support the optional [PCIe MMPT ECN], CXL components need to implement additional Config Space and Memory Space registers. Also, firmware changes are required to support the new interfaces and commands while maintaining compatibility with CXL software that doesn't support the [PCIe MMPT ECN].

**5. Analysis of the Software Implications**

To support the optional [PCIe MMPT ECN], CXL software needs to be updated to implement the management interfaces in the [PCIe MMPT ECN] while maintaining compatibility with CXL components that don't support the [PCIe MMPT ECN].

**6. Analysis of the Compliance and Test Implications**

While the [PCIe MMPT ECN] is optional, compliance tests could be added to ensure backwards compatibility for CXL components implementing the [PCIe MMPT ECN].

**Part II - Detailed Description of the changes**

*Add the following to Section 1.3 Reference Documents:*

| Document | Chapter Reference | Document No./Location |
|---|---|---|
| Management Message Passthrough via MMIO Mailbox (MMPT) ECN to PCI Express Specification Revision 6.1 | N/A | https://members.pcisig.com/wg/PCI-SIG/document/20109 |

*Update Section 8.2.8 with the following changes:*

### 8.2.8 CXL Device Register Interface

CXL device registers are mapped in memory space allocated via a standard PCIe BAR. The entry in the Register Locator DVSEC structure (see Section 8.1.9) with Register Identifier = 03h 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 that 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.

An MLD shall implement one instance of CXL Device registers in the MMIO space of each applicable LD.
Figure 8-12 CXL Device Registers

...

IMPLEMENTATION NOTE

**Compatibility with PCIe MMIO Capabilities (MCAP) Register Block**

CXL components are expected to transition to the PCIe standard MCAP Register Block to enumerate CXL defined capabilities. When doing so, CXL components are required to maintain compatibility with legacy CXL software as portrayed in Figure XX.

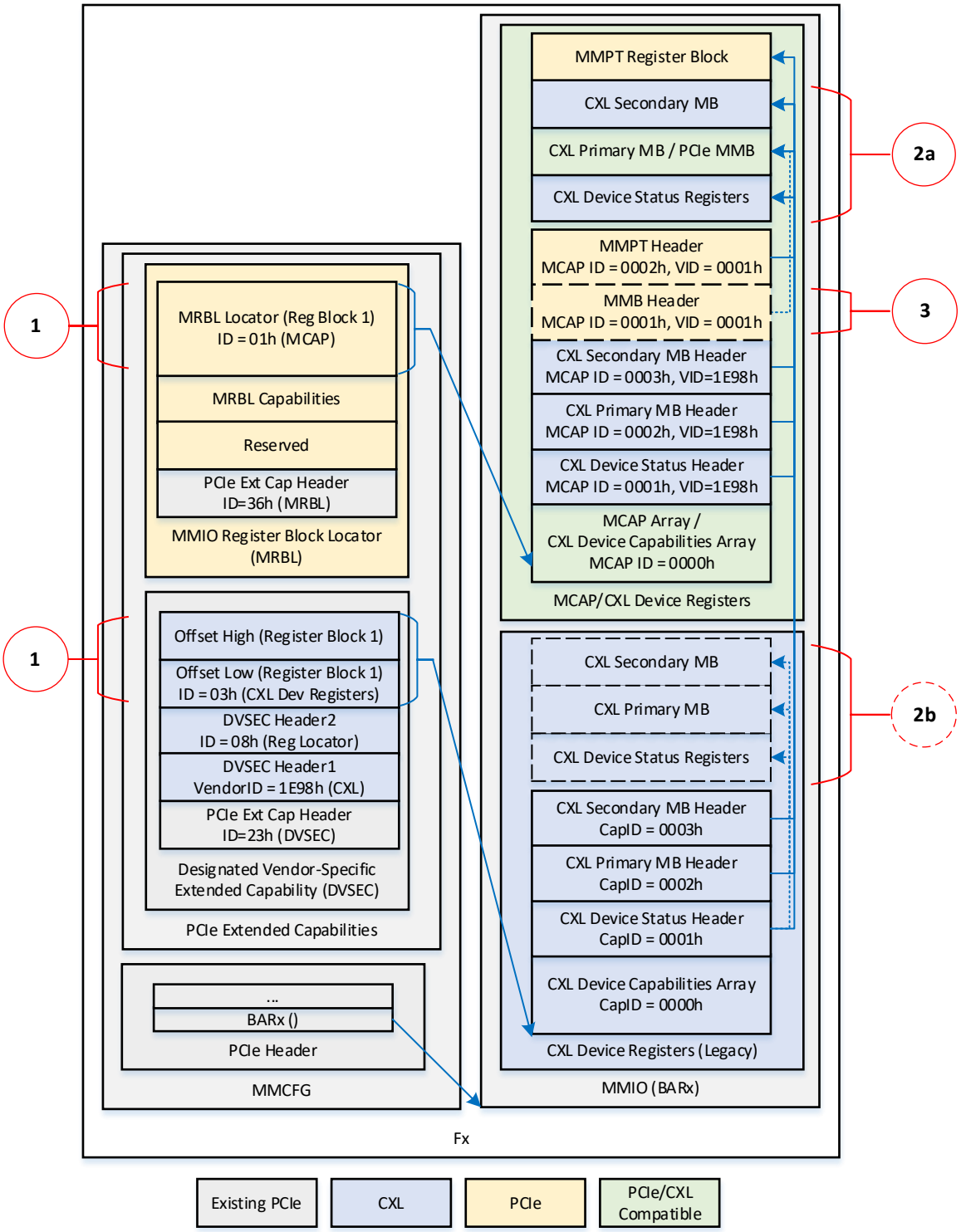


*Figure xx PCIe MCAP / CXL Compatibility*

1. CXL components are required to advertise CXL defined capabilities in both the legacy CXL Device Capabilities Array and the PCIe MCAP Array. Legacy CXL software discovers the CXL Device Capabilities Array using the entry in the CXL Register Locator DVSEC with Register Identifier = 03h (CXL Device Registers). Updated software discovers the PCIe MCAP Array using the entry in the PCIe MMIO Register Block Locator (MRBL) Extended Capability with Register Block Identifier = 01h (MMIO Capabilities). Updated SW should check the PCIe MCAP Array first and only fallback to the legacy CXL Device Capabilities Array if it's not supported.
2. While CXL defined capabilities must be advertised in both the legacy CXL Device Capabilities Array and the PCIe MCAP Array, CXL components may choose to implement the register

structures for CXL defined capabilities in one of the following ways. In either case, legacy software locates the register structure from the offset specified in the header in the legacy CXL Device Capabilities Array. Updated software locates the register structure from the offset specified in the header in the PCIe MCAP Array.

a. CXL components may alias the location of CXL defined capabilities specified in the legacy CXL Device Capabilities Array and the PCIe MCAP Array to a single instance of the register structure. In this case, the CXL Device Capabilities Array, the PCIe MCAP Array, and the register structures are required to be located in the same PCIe BAR and both the CXL Device Capabilities Array and the PCIe MCAP Array must be located below the register structures.

b. CXL components may choose to implement two instances of the register structures for CXL defined capabilities; one located from the offset specified in the header in the CXL Device Capabilities Array and one located from the offset specified in the header in the PCIe MCAP Array. Updated software is required to use the instance of the register structure located via the PCIe MCAP Array.

3. CXL components that support both the CXL primary mailbox and the PCIe MMB are required to alias the PCIe MMB (Vendor ID = 0001h and MCAP ID = 0001h) to the CXL Primary Mailbox registers. Refer to Section 9.20 for complete details.

#### 8.2.8.1 CXL Device Capabilities Array Register (Offset 00h)

IMPLEMENTATION NOTE

CXL components are recommended to transition to the equivalent PCIe MCAP Array Register as defined in [PCIe].

| Bits | Attributes | Description |
|---|---|---|
| 15:0 | ~~RO~~ - | **Capability ID:** ~~...~~ See equivalent MCAP ID field in the MCAP Array Register[1]. |
| 23:16 | ~~RO~~ - | **Version:** ~~...~~ See equivalent MCAP Array Version field in the MCAP Array Register[1]. |
| 27:24 | ~~RO~~ - | **Type:** ~~...~~ See equivalent MCAP Type field in the MCAP Array Register[1]. For CXL, bit 27 is fixed to 0 and bits 26:24 are defined in Table XX CXL defined type specific capabilities. |
| 31:28 | ~~RO~~ - | ~~**Reserved**~~Refer to [PCIe] for the definition of this field. |
| 47:32 | ~~RO~~ - | **Capabilities Count:** ~~...~~ See equivalent MCAP Count field in the MCAP Array Register[1]. |
| 127:48 | ~~RO~~ - | ~~**Reserved**~~Refer to [PCIe] for the definition of this field. |

[1]Refer to [PCIe] for the definition of this field.

*Table XX CXL Defined Type-Specific Capabilities*

| Type | Description |
|---|---|
| 0h | ~~The type is inferred from the PCI Class code. If the PCI Class code is not associated with a type defined by this specification, no type-specific capabilities are present.~~ Reserved for PCIe[1]. |
| 1h | Memory Device Capabilities (see Section 8.2.8.5). |
| 2h | Switch Mailbox CCI Capabilities (see Section 8.2.8.6). |
| 3-7h | Reserved |

[1]Refer to [PCIe] for the definition of this value.

#### 8.2.8.2 CXL Device Capability Header Register (Offset: Varies)

Each capability in the CXL device capabilities array is described by a CXL Device Capability Header register that identifies the specific capability and points to the capability register structure in register space.

IMPLEMENTATION NOTE

CXL components are recommended to transition to the equivalent PCIe MCAP Header Register as defined in [PCIe].

| Bits | Attributes | Description |
|---|---|---|
| 15:0 | ~~RO~~ - | **Capability ID:** ~~...~~ See equivalent MCAP ID field in the MCAP Header Register[1]. See Section 8.2.8.2.1 for the list of CXL defined capability identifiers. |
| 23:16 | ~~RO~~ - | **Version:** ~~...~~See equivalent MCAP Version field in the MCAP Header Register[1]. |
| 31:24 | ~~RO~~ - | ~~**Reserved**~~Refer to [PCIe] for the definition of this field. |
| 63:32 | ~~RO~~ - | **Offset:** ~~...~~ See equivalent MCAP Offset field in the MCAP Header Register[1]. |
| 95:64 | ~~RO~~ - | **Length:** ~~...~~ See equivalent MCAP Length field in the MCAP Header Register[1]. |
| 127: 96 | ~~RO~~ - | ~~**Reserved**~~Refer to [PCIe] for the definition of this field. |

[1]Refer to [PCIe] for the definition of this field.

##### 8.2.8.2.1 CXL Device Capabilities

CXL defined device capability register structures are identified by a 2-byte identifier.

- CXL defined capability identifiers 0000h-3FFFh describe generic CXL device capabilities as specified in ~~the~~ table ~~below~~ XX.
- CXL defined capability identifiers 4000h-7FFFh describe type-specific capabilities associated with the type specified in the CXL Device Capabilities Array register (see Section 8.2.8.1).
- Capability identifiers 8000h-FFFFh describe vendor specific capabilities.

CXL defined capability identifiers 0000h-3FFFh that are not specified in this table are reserved.

*Table XX* CXL Defined Capability Identifiers (Vendor ID = 1E98h or 0000h)

...

IMPLEMENTATION NOTE

CXL components are permitted to set the PCIe MCAP Vendor ID to 0000h for CXL defined capabilities. 0000h is a PCI-SIG reserved value for legacy CXL compatibility. However, it is strongly recommended to use the CXL Vendor ID (1E98h) to identify CXL defined capabilities.

#### 8.2.8.4 Mailbox Registers (Offset Varies)

CXL defined extensions to the PCIe standard MMIO Mailbox Capability (MMB) Registers are described in this section. Refer to the MMIO Mailbox Capability (MMB) Section of the [PCIe MMPT ECN] for the definition of fields not listed.

~~The mailbox registers provide the ability to issue a command to the device.~~ There are two types of CXL mailboxes provided through the device's register interface: primary and secondary. Each mailbox

represents a unique CCI instance in the device and the properties of each instance are defined in Section 9.1.1. The secondary mailbox does not support background operations. The status of a background operation issued to a device's primary mailbox can be retrieved from the Background Command Status register, as detailed in Section 8.2.8.4.7.

The register Interface for both types of mailboxes is the same and is described in the MMB Registers section of [PCIe]. CXL defined extensions to the MMB Registers are called out in this section. The difference between the two types of mailboxes is their intended use and commands allowed. Details on these differences are described in Section 8.2.8.4.1 and Section 8.2.8.4.2.

~~The primary and the secondary mailbox interfaces shall only be used in a single-threaded manner. It is software's responsibility to avoid simultaneous, uncoordinated access to the mailbox registers using techniques such as locking.~~

~~The mailbox command timeout is 2 seconds.~~ Commands that require a longer execution time than the MMB command timeout shall be completed asynchronously in the background. Only one command can be executed in the background at a time. The status of a background command can be retrieved from the Background Command Status register. Background commands do not continue to execute across Conventional Resets. For devices with multiple mailboxes, only the primary mailbox shall be used to issue background commands.

~~Devices may support sending MSI/MSI-X interrupts to indicate command status. Support for mailbox interrupts is enumerated in the Mailbox Capabilities register and enabled in the Mailbox Control register. Mailbox interrupts are only supported on the primary mailbox.~~
~~Unless specified otherwise in the field definitions for the mailbox registers below, each field is present in version 1 and later of these structures.~~ The device shall report the version of these structures in the Version field of the CXL Device Capability Header register.

~~The flow for executing a command is described below. The term "Caller" represents the entity submitting the command:~~
~~1. Caller reads MB Control register to verify doorbell is cleared.~~
~~2. Caller writes Command register.~~
~~3. Caller writes Command Payload registers if input payload is non-empty.~~
~~4. Caller writes MB Control register to set doorbell.~~
~~5. Caller either polls for doorbell to be cleared or waits for interrupt if configured.~~
~~6. Caller reads MB Status register to fetch Return code.~~
~~7. If command is successful, Caller reads Command register to get Payload Length.~~
~~8. If output payload is non-empty, Caller reads Command Payload registers.~~

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

...

#### 8.2.8.4.2 Attributes of the Secondary Mailbox

The secondary mailbox, if implemented by a device, supports only a subset of the commands described in Section 8.2.9. The Command Effects Log shall specify which CXL defined commands are allowed on the secondary mailbox, and all other commands shall return the error Unsupported Mailbox or CCI. The secondary mailbox does not support mailbox completion interrupts. Therefore, any fields related to command interrupts shall be set to 0 on the Secondary MB. Implementation of the secondary mailbox is optional.

#### 8.2.8.4.3 Mailbox Capabilities Register (Mailbox Registers Capability Offset + 00h)

The Mailbox Capabilities Register extends the definition of the PCIe MMB Capabilities Register with the following CXL defined fields.

| Bits | Attributes | Description |
|---|---|---|
| 4:0 | ~~RO~~ - | **Payload Size:** ~~...~~ See equivalent MMB Payload Registers Size field in the MMB Capabilities Register[1]. |
| 5 | ~~RO~~ - | **MB Doorbell Interrupt Capable:** ~~...~~ See equivalent Command Ready Interrupt Capable field in the MMB Capabilities Register[1]. |
| 6 | RO | **Background Command Complete Interrupt Capable:** When set, indicates the device supports signaling an MSI/MSI-X interrupt when a command completes in the background. Only valid for the primary mailbox. This bit shall be 0 for the secondary mailbox. |
| 10:7 | ~~RO~~ - | **Interrupt Message Number:** ~~...~~ See equivalent Interrupt Message Number field in the MMB Capabilities Register[1]. |
| 18:11 | ~~RO~~ - | **Mailbox Ready Time:** ~~...~~ See equivalent MMB Ready Time field in the MMB Capabilities Register[1]. |
| 22:19 | ~~RO~~ - | **Type:** ~~...~~ See equivalent Type field in the MMB Capabilities Register[1]. For CXL, bit 22 is fixed to 0 and bits 21:19 are defined in Table XX CXL Defined Mailbox Type Identifiers. |
| 31:23 | ~~RO~~ - | ~~**Reserved**~~Refer to [PCIe] for the definition of this field. |

[1]Refer to [PCIe] for the definition of this field.

Table XX – CXL Defined Mailbox Type Identifiers.

| Type | Description |
|---|---|
| 0h | Reserved for PCIe[1]. |
| 1h | Memory Device Commands (see Section 8.2.9.9). |
| 2h | FM API Commands (see Section 8.2.9.10). |
| 3-7h | Reserved |

[1]Refer to [PCIe] for the definition of this type.

#### 8.2.8.4.4 Mailbox Control Register (Mailbox Registers Capability Offset + 04h)

The Mailbox Control Register extends the definition of the PCIe MMB Control Register with the following CXL defined fields.

| Bits | Attributes | Description |
|---|---|---|
| 0 | ~~RW/RO~~ - | **Doorbell:** ~~...~~ See equivalent Doorbell field in the MMB Control Register[1]. |
| 1 | ~~RW/RO~~ - | **MB Doorbell Interrupt:** ~~...~~ See equivalent Command Ready Interrupt Enable field in the MMB Control Register[1]. |
| 2 | RW/RO | **Background Command Complete Interrupt:** If background command complete interrupts are supported on this mailbox, this register is set by the caller to |

| | | |
|---|---|---|
| | | enable signaling an interrupt when the command completes in the background. Ignored if the command is not a background command. RO when the doorbell is set. Ignored if background command complete interrupts are not supported on this mailbox instance (Background Command Complete Interrupt Capable = 0 in the Mailbox Capabilities register).<br>• 0 = Disabled<br>• 1 = Enabled |
| 31:3 | ~~RsvdP~~ - | ~~**Reserved**~~Refer to [PCIe] for the definition of this field. |

[1]Refer to [PCIe] for the definition of this field.

#### 8.2.8.4.5 Command Register (Mailbox Registers Capability Offset + 08h)

The Command Register is equivalent to the definition of the PCIe MMB Control Register with the following CXL defined fields.

~~This register shall only be used by the caller when the doorbell in the Mailbox Control register is cleared.~~

| Bits | Attributes | Description |
|---|---|---|
| 15:0 | ~~RW~~ - | **Command Opcode:** ~~...~~ See equivalent MMB Command Opcode field in the MMB Command Register[1]. |
| 36:16 | ~~RW~~ - | **Payload Length:** ~~...~~ See equivalent MMB Payload Length field in the MMB Command Register[1]. |
| 63:37 | ~~RsvdP~~ - | ~~**Reserved**~~Refer to [PCIe] for the definition of this field. |

[1]Refer to [PCIe] for the definition of this field.

IMPLEMENTATION NOTE

CXL components shall interpret the PCIe MMB Command Opcode Vendor ID = 1E98h or 0000h with CXL defined commands. 0000h is a PCI-SIG reserved value for legacy CXL compatibility. However, it is strongly recommended for callers to use the CXL Vendor ID (1E98h) to identify CXL defined commands.

##### 8.2.8.4.5.1 Command Return Codes

~~In general, retries are not recommended for commands that return an error except when indicated in the return code definition.~~

CXL defined command return codes are only interpreted for CXL defined commands.

Table 8-34 CXL defined Command Return Codes (Vendor ID = 1E98h or 0000h)

...

#### 8.2.8.4.6 Mailbox Status Register (Mailbox Registers Capability Offset + 10h)

The Mailbox Status Register extends the definition of the PCIe MMB Status Register with the following CXL defined fields.

| Bits | Attributes | Description |
|---|---|---|
| 0 | RO | **Background Operation:** When set, the device is executing a command in the background. Only one command can be executing in the background; therefore, additional background commands shall be rejected with the busy return code. Referto the Background Command Status register to retrieve the |

| | | |
|---|---|---|
| | | status of the background command. Only valid for the primary mailbox. This bit shall be 0 for the secondary mailbox. |
| 31:1 | ~~RO~~ - | ~~**Reserved**~~Refer to [PCIe] for the definition of this field. |
| 47:32 | RO | **Return Code:** ~~...~~ See equivalent MMB Return Code field in the MMB Status Register[1]. |
| 31:3 | RO | **Vendor Specific Extended Status:** ~~...~~ See equivalent Vendor-Specific Extended Status field in the MMB Status Register[1]. |

[1]Refer to [PCIe] for the definition of this field.

#### 8.2.8.4.7 Background Command Status Register (Mailbox Registers Capability Offset + 18h)

Reports information about the last command executed in the background since the last Conventional Reset. Zeroed if no background command status is available. Valid only for the primary mailbox. This register shall be zeroed on the secondary mailbox.

| Bits | Attributes | Description |
|---|---|---|
| 15:0 | RO | **Command Opcode:** ... |
| 22:16 | RO | **Percentage Complete:** ... |
| 31:23 | RO | **Reserved:** ... |
| 47:32 | RO | **Return Code:** ... |
| 63:48 | RO | **Vendor Specific Extended Status:** ... |

#### 8.2.8.4.8 Command Payload Registers (Mailbox Registers Capability Offset + 20h)

The Command Payload Registers are equivalent to the definition of the PCIe MMB Payload Registers. Refer to [PCIe] for the definition. See Section 8.2.9 for the format of the payload data for each CXL defined command.

~~These registers shall only be used by the caller when the doorbell in the Mailbox Control register is cleared.~~

~~...~~

### 8.2.8.5 Memory Device Capabilities

This section describes...

*Table 8-35* CXL Defined Memory Device Capability Identifiers (Vendor ID = 1E98h or 0000h)

...

IMPLEMENTATION NOTE

CXL components are permitted to set the PCIe MCAP Vendor ID to 0000h for CXL defined memory device capabilities. 0000h is a PCI-SIG reserved value for legacy CXL compatibility. However, it is strongly recommended to use the CXL Vendor ID (1E98h) to identify CXL defined memory device capabilities.

### 8.2.8.5 Switch Device Capabilities

This section describes...

*Table 8-36* CXL Defined Switch Mailbox CCI Capability Identifiers (Vendor ID = 1E98h or 0000h)

...

IMPLEMENTATION NOTE
CXL components are permitted to set the PCIe MCAP Vendor ID to 0000h for CXL defined switch device capabilities. 0000h is a PCI-SIG reserved value for legacy CXL compatibility. However, it is strongly recommended to use the CXL Vendor ID (1E98h) to identify CXL defined switch device capabilities.

*Update Section 8.2.9 with the following changes:*

### 8.2.9 Component Command Interface

CXL defined component commands are identified by a 2-byte Opcode and the CXL Vendor ID (1E98h or 0000h).

...

*Table 8-37* CXL Defined Generic Component Command Opcodes (Vendor ID = 1E98h or 0000h)

...

IMPLEMENTATION NOTE
CXL components shall interpret the PCIe MMB Command Opcode Vendor ID = 1E98h or 0000h with CXL defined commands. 0000h is a PCI-SIG reserved value for legacy CXL compatibility. However, it is strongly recommended for callers to use the CXL Vendor ID (1E98h) to identify CXL defined commands.

#### 8.2.9.5.2.1 Command Effects Log (CEL)

The Command Effects Log (CEL) is a variable-length log page that reports each CXL defined command supported by the CCI that was queried (with the exception of the secondary mailbox, as described in Section 8.2.8.4.2) and the effect each command may have on the device subsystem.

...

### 8.2.9.9 Memory Device Command Sets

...

*Table 8-126* CXL Defined Memory Device Command Opcodes (Vendor ID = 1E98h or 0000h)

...

IMPLEMENTATION NOTE
CXL components shall interpret the PCIe MMB Command Opcode Vendor ID = 1E98h or 0000h with CXL defined commands. 0000h is a PCI-SIG reserved value for legacy CXL compatibility. However, it is strongly recommended for callers to use the CXL Vendor ID (1E98h) to identify CXL defined commands.

### 8.2.9.10 FM API Commands

...

*Table 8-215* CXL Defined FM API Command Opcodes (Vendor ID = 1E98h or 0000h)

...

IMPLEMENTATION NOTE
CXL components shall interpret the PCIe MMB Command Opcode Vendor ID = 1E98h or 0000h with CXL defined commands. 0000h is a PCI-SIG reserved value for legacy CXL compatibility. However, it is strongly recommended for callers to use the CXL Vendor ID (1E98h) to identify CXL defined commands.

*Update Section 9.20 with the following changes:*

**9.20 Component Command Interface**

Runtime management of CXL components is facilitated by a Component Command Interface (CCI). A CCI represents a command target that is used to process management and configuration commands that are issued to the component. Table 8-37, Table 8-126, and Table 8-215 define the commands that a CCI can support. A component can implement multiple CCIs of varying types that operate independently of one another and that have a uniquely defined list of supported commands. There are 2 types of CCIs:

- CXL Mailbox Registers: A component can expose up to 2 CXL mailboxes through its Mailbox registers, as defined in Section 8.2.8.4. Each mailbox represents a unique CCI instance.
- MCTP-based CCIs: Components with MCTP-capable interconnects can expose up to 1 CCI per interconnect. There is a 1:1 relationship between the component's MCTP based CCIs and MCTP-capable interconnects. Transfer of commands via MCTP uses the transport protocol defined in Section 7.6.3.

All CCIs shall comply with the properties described in Section 9.20.1.

IMPLEMENTATION NOTE

The CXL mailbox is derived from the PCIe standard MMIO Mailbox Capability (MMB) with extensions defined in Section 8.2.8.4 for supporting CXL defined commands. Therefore, the CXL mailbox may also support PCI-SIG defined commands (MMB Command Opcode Vendor ID = 0001h) or commands defined by other entities. However, non-CXL defined commands are not reported in the CXL CEL and discovery of those commands is outside of the scope of this specification.

CXL components that need to be compatible with non-CXL aware software may advertise both the CXL Primary Mailbox (Vendor ID = 1E98h or 0000h, ID = 0002h) and the PCIe MMB (Vendor ID = 0001h, ID = 0001h). However, they are required to alias the PCIe MMB header to the CXL Primary Mailbox registers. Refer to Section 8.2.8, Figure XX. CXL components that do not need to be compatible with non-CXL aware software should only advertise the CXL Primary Mailbox and not the PCIe MMB.