



1

2 **Management Component Transport Protocol**  
3 **(MCTP) PCIe® VDM Transport Binding**  
4 **Specification**

5 **Version: 1.4.0**

6 **Document Identifier: DSP0238**

7 **Date: 2026-01-16**

8 **Version History: <https://www.dmtf.org/dsp/DSP0238>**

9 **Supersedes: 1.3.1**

10 **Document Class: Normative**

11 **Document Status: Published**

12 **Document Language: en-US**

13 Copyright Notice

14 Copyright © 2009, 2014, 2018, 2021, 2023–2026 DMTF. All rights reserved.

15 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems  
16 management and interoperability. Members and non-members may reproduce DMTF specifications and  
17 documents for uses consistent with this purpose, provided that correct attribution is given. As DMTF  
18 specifications may be revised from time to time, the particular version and release date should always be  
19 noted.

20 Implementation of certain elements of this standard or proposed standard may be subject to third-party  
21 patent rights, including provisional patent rights (herein “patent rights”). DMTF makes no representations  
22 to users of the standard as to the existence of such rights and is not responsible to recognize, disclose, or  
23 identify any or all such third-party patent right owners or claimants, nor for any incomplete or inaccurate  
24 identification or disclosure of such rights, owners, or claimants. DMTF shall have no liability to any party,  
25 in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or  
26 identify any such third-party patent rights, or for such party’s reliance on the standard or incorporation  
27 thereof in its products, protocols, or testing procedures. DMTF shall have no liability to any party  
28 implementing such standards, whether such implementation is foreseeable or not, nor to any patent  
29 owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is  
30 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  
31 implementing the standard from any and all claims of infringement by a patent owner for such  
32 implementations.

33 For information about patents held by third parties which have notified DMTF that, in their opinion, such  
34 patents may relate to or impact implementations of DMTF standards, visit  
35 <https://www.dmtf.org/about/policies/disclosures>.

36 CXL® and Compute Express Link® are registered trademarks of the Compute Express Link Consortium.  
37 PCI-SIG®, PCI Express®, and PCIe® are registered trademarks or service marks of PCI-SIG. All other  
38 marks and brands are the property of their respective owners.

39 This document’s normative language is English. Translation into other languages is permitted.

40

# CONTENTS

41	Foreword .....	5
42	Introduction .....	6
43	1 Scope .....	7
44	2 Normative references .....	7
45	3 Terms and definitions .....	8
46	3.1 Destination Physical Address .....	8
47	3.2 Dword .....	8
48	3.3 Flit Mode .....	8
49	3.4 MCTP PCIe Endpoint .....	8
50	3.5 PCIe Segment .....	8
51	3.6 Target Physical Address .....	8
52	4 Symbols and abbreviated terms .....	9
53	4.1 PCIe® .....	9
54	4.2 VDM .....	9
55	4.3 CXL® .....	9
56	4.4 FM .....	9
57	4.5 NFM .....	9
58	4.6 IDE .....	9
59	4.7 OHC .....	9
60	4.8 TLP .....	9
61	5 Conventions .....	10
62	5.1 Reserved and unassigned values .....	10
63	5.2 Byte ordering .....	10
64	6 MCTP over PCI Express VDM transport .....	10
65	6.1 Overview .....	10
66	6.2 Packet formats .....	10
67	6.3 Supported media .....	16
68	6.4 Physical address format for MCTP control messages .....	17
69	6.5 Message routing .....	17
70	6.6 Bus owner address .....	21
71	6.7 Bus and Segment address assignment for PCIe .....	21
72	6.8 Host dependencies .....	22
73	6.9 Discovery Notify message use for PCIe .....	23
74	6.10 MCTP over PCIe endpoint discovery .....	24
75	6.11 MCTP messages timing requirements .....	28
76	ANNEX A (informative) Notations and conventions .....	30
77	ANNEX B (informative) Change log .....	31
78		

79 **Figures**

80	Figure 1 – MCTP over PCI Express Vendor Defined Message (VDM) packet format (Non-Flit Mode) .....	11
81	Figure 2 – MCTP over PCI Express Vendor Defined Message (VDM) packet format (Flit Mode within a	
82	Segment without IDE) .....	13
83	Figure 3 – MCTP over PCI Express Vendor Defined Message (VDM) packet format (Flit Mode across	
84	segments and/or with IDE).....	14
85	Figure 4 – Flit Mode to Non-Flit Mode routing options.....	19
86	Figure 5 – Bus owner and Root Complex co-located .....	23
87	Figure 6 – Bus owner and Root Complex separated.....	23
88	Figure 7 – Example Flow of operations for full MCTP Discovery over PCIe .....	26
89	Figure 8 – Flow of operations for Partial Endpoint Discovery.....	27
90		

91 **Tables**

92	Table 1 – PCI Express medium-specific MCTP packet fields.....	11
93	Table 2 – PCI Express medium-specific MCTP Packet Fields in Flit Mode .....	15
94	Table 3 – Supported media.....	16
95	Table 4 – Physical address format (Non-Flit Mode) .....	17
96	Table 5 – Physical address format (Flit Mode) .....	17
97	Table 6 – Address Translations when transitioning between Flit Mode and Non-Flit Mode hierarchies....	19
98	Table 7 – Address used for routing examples .....	21
99	Table 8 – Timing specifications for MCTP messages on PCIe VDM.....	28
100		

101

## Foreword

102 The *Management Component Transport Protocol (MCTP) PCIe® VDM Transport Binding Specification*  
103 (DSP0238) was prepared by the PMCI Working Group.

104 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems  
105 management and interoperability.

### 106 Acknowledgments

107 DMTF acknowledges the following individuals for their contributions to this document:

#### 108 Editors:

- 109 • Hemal Shah – Broadcom Inc.
- 110 • Tom Slaight – Intel Corporation
- 111 • Eliel Louzoun – Intel Corporation

#### 112 Contributors:

- 113 • Austin Bolen – Dell Technologies
- 114 • Patrick Caporale – Lenovo
- 115 • Michał Duzinkiewicz – Intel Corporation
- 116 • Samer El-Haj-Mahmoud - ARM
- 117 • Steve Glaser – NVIDIA Corporation
- 118 • Brett Henning – Broadcom Inc.
- 119 • Yuval Itkin – NVIDIA Corporation
- 120 • Janusz Jurski – Intel Corporation
- 121 • Jose Marinho – ARM
- 122 • Mariusz Oriol – Intel Corporation
- 123 • Patrick Schoeller – Hewlett Packard Enterprise
- 124 • Bob Stevens – Dell Technologies

125

## Introduction

126 The Management Component Transport Protocol (MCTP) over PCIe VDM transport binding defines a  
127 transport binding for facilitating communication between platform management subsystem components  
128 (e.g., management controllers, management devices) over PCIe.

129 The [MCTP Base Specification](#) describes the protocol and commands used for communication within and  
130 initialization of an MCTP network. The MCTP over PCIe VDM transport binding definition in this  
131 specification includes a packet format, physical address format, message routing, and discovery  
132 mechanisms for MCTP over PCIe VDM communications.

## 133 1 Scope

134 This document provides the specifications for the Management Component Transport Protocol (MCTP)  
135 transport binding using PCIe Vendor Defined Messages (VDMs).

## 136 2 Normative references

137 The following referenced documents are indispensable for the application of this document. For dated  
138 references, only the edition cited applies. For undated references, the latest edition of the referenced  
139 document (including any amendments) applies.

140 CXL Consortium, *Compute Express Link (CXL) Specification Revision 1.0*, March 2019,  
141 <https://www.computeexpresslink.org>

142 CXL Consortium, *Compute Express Link (CXL) Specification Revision 1.1*, June 2019,  
143 <https://www.computeexpresslink.org>

144 CXL Consortium, *Compute Express Link (CXL) Specification Revision 2.0*, October 26, 2020,  
145 <https://www.computeexpresslink.org>

146 CXL Consortium, *Compute Express Link (CXL) Specification Revision 3.2*, October 2, 2024,  
147 <https://www.computeexpresslink.org>

148 CXL Consortium, *Compute Express Link (CXL) Specification Revision 4.0*, August 13, 2025,  
149 <https://www.computeexpresslink.org>

150 DMTF DSP0236, *Management Component Transport Protocol (MCTP) Base Specification 1.3*,  
151 [https://www.dmtf.org/sites/default/files/standards/documents/DSP0236\\_1.3.pdf](https://www.dmtf.org/sites/default/files/standards/documents/DSP0236_1.3.pdf)

152 DMTF DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes 1.9*,  
153 [https://www.dmtf.org/sites/default/files/standards/documents/DSP0239\\_1.9.pdf](https://www.dmtf.org/sites/default/files/standards/documents/DSP0239_1.9.pdf)

154 ISO/IEC Directives, Part 2, *Principles and rules for the structure and drafting of ISO and IEC documents*,  
155 <https://www.iso.org/sites/directives/current/part2/index.xhtml>

156 PCI-SIG, *PCI Express Base Specification Revision 1.1*, March 8, 2005,  
157 <https://www.pcisig.com/specifications/>

158 PCI-SIG, *PCI Express Base Specification Revision 2.0*, December 20, 2006,  
159 <https://www.pcisig.com/specifications/>

160 PCI-SIG, *PCI Express Base Specification Revision 2.1*, March 4, 2009,  
161 <https://www.pcisig.com/specifications/>

162 PCI-SIG, *PCI Express Base Specification Revision 3.0*, November 10, 2010,  
163 <https://www.pcisig.com/specifications/>

164 PCI-SIG, *PCI Express Base Specification Revision 3.1a*, December 7, 2015,  
165 <https://www.pcisig.com/specifications/>

166 PCI-SIG, *PCI Express Base Specification Revision 4.0*, October 5, 2017,  
167 <https://www.pcisig.com/specifications/>

168 PCI-SIG, *PCI Express Base Specification Revision 5.0*, May 28, 2019,  
169 <https://www.pcisig.com/specifications/>

170 PCI-SIG, *PCI Express Base Specification Revision 6.2*, January 25, 2024,  
171 <https://www.pcisig.com/specifications/>

172 PCI-SIG, *PCI Express Base Specification Revision 7.0*, June 11, 2025,  
173 <https://www.pcisig.com/specifications/>

### 174 3 Terms and definitions

175 In this document, some terms have a specific meaning beyond the normal English meaning. Those terms  
176 are defined in this clause.

177 The terms "shall" ("required"), "shall not", "should" ("recommended"), "should not" ("not recommended"),  
178 "may", "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described  
179 in [ISO/IEC Directives, Part 2](#), Clause 7. The terms in parentheses are alternatives for the preceding term,  
180 for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that  
181 [ISO/IEC Directives, Part 2](#), Clause 7 specifies additional alternatives. Occurrences of such additional  
182 alternatives shall be interpreted in their normal English meaning.

183 The terms "clause", "subclause", "paragraph", and "annex" in this document are to be interpreted as  
184 described in [ISO/IEC Directives, Part 2](#), Clause 6.

185 The terms "normative" and "informative" in this document are to be interpreted as described in [ISO/IEC](#)  
186 [Directives, Part 2](#), Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do  
187 not contain normative content. Notes and examples are always informative elements.

188 Refer to [DSP0236](#) for terms and definitions that are used across the MCTP specifications. For the  
189 purposes of this document, the following additional terms and definitions apply.

#### 190 3.1 Destination Physical Address

191 the Physical address reflected in the Requester ID field (Non-Flit Mode) or Requester ID and Requester  
192 Segment (Flit Mode) fields of the TLP

#### 193 3.2 Dword

194 a 32-bit field

#### 195 3.3 Flit Mode

196 a mode defined in the PCI Express Base Specification Revision 6.x introducing a new link layer and  
197 transaction layer for PCI Express where a TLP header is composed of a 3 to 7 dword TLP Header Base  
198 followed by 0 to 7 additional dwords of OHCs (Orthogonal Header Content)

#### 199 3.4 MCTP PCIe Endpoint

200 a PCIe endpoint on which MCTP PCIe VDM communication is supported

#### 201 3.5 PCIe Segment

202 a PCI Express I/O interconnect topology, in which the Requester IDs must be unique

#### 203 3.6 Target Physical Address

204 The Physical address reflected in the Target ID field (Non-Flit Mode) or Target ID and Target Segment  
205 (Flit Mode) fields of the TLP

## 206 **4 Symbols and abbreviated terms**

207 Refer to [DSP0236](#) for symbols and abbreviated terms that are used across the MCTP specifications. The  
208 following symbols and abbreviations are used in this document.

### 209 **4.1 PCIe®**

210 PCI Express®

### 211 **4.2 VDM**

212 Vendor Defined Message

### 213 **4.3 CXL®**

214 Compute Express Link®

### 215 **4.4 FM**

216 Flit Mode

### 217 **4.5 NFM**

218 Non-Flit Mode

### 219 **4.6 IDE**

220 Integrity and Data Encryption

### 221 **4.7 OHC**

222 Orthogonal Header Content

### 223 **4.8 TLP**

224 Transaction Layer Packet

## 225 5 Conventions

226 The conventions described in the following clauses apply to this specification.

### 227 5.1 Reserved and unassigned values

228 Unless otherwise specified, any reserved, unspecified, or unassigned values in enumerations or other  
229 numeric ranges are reserved for future definition by DMTF.

230 Unless otherwise specified, numeric or bit fields that are designated as reserved shall be written as 0  
231 (zero) and ignored when read.

### 232 5.2 Byte ordering

233 Unless otherwise specified, byte ordering of multi-byte numeric fields or bit fields is "Big Endian" (that is,  
234 the lower byte offset holds the most significant byte, and higher offsets hold lesser significant bytes).

## 235 6 MCTP over PCI Express VDM transport

### 236 6.1 Overview

237 This document defines the medium-specific transport binding for transferring MCTP packets between  
238 endpoints on PCI Express using PCIe Vendor Defined Messages (VDMs).

239 An MCTP over PCIe VDM compliant PCIe device shall support MCTP over PCIe VDM communications on  
240 at least one PCIe Physical Function (PF) of the device. If an MCTP over PCIe VDM compliant PCI device  
241 supports MCTP over PCIe VDM communications on more than one PCIe function, then MCTP over PCIe  
242 VDM communication on each function shall be independent from MCTP over PCIe VDM communications  
243 on other PCIe functions.

244 The MCTP over PCI Express (PCIe) VDM transport binding transfers MCTP messages using PCIe Type  
245 1 VDMs with data. MCTP messages use the MCTP VDM code value (0000b) stored in the PCIe TAG  
246 field that uniquely differentiates MCTP messages from other DMTF VDMs.

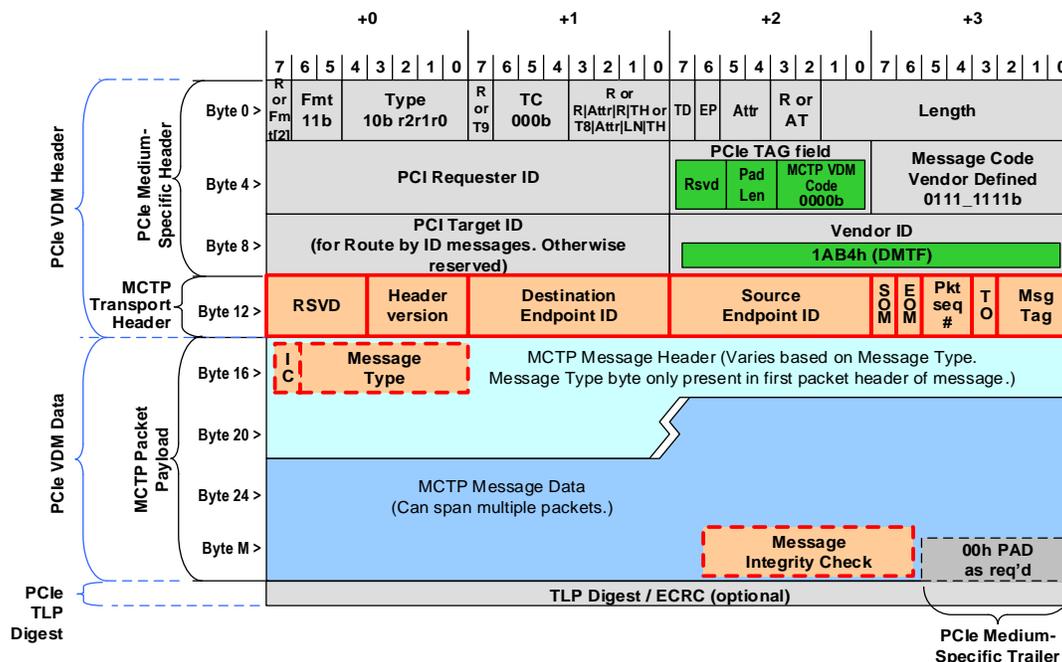
247 Any MCTP baseline transmission unit for MCTP PCIe VDM communication shall be dword aligned.

248 The handling of transactions with parameters not matching the values described below is out of scope for  
249 this specification. The receiver may choose to accept those or ignore them.

### 250 6.2 Packet formats

#### 251 6.2.1 Non-Flit Mode

252 Figure 1 shows the encapsulation of MCTP packet fields within a PCIe VDM in Non-Flit Mode.



253

254  
255

**Figure 1 – MCTP over PCI Express Vendor Defined Message (VDM) packet format (Non-Flit Mode)**

256 The fields labeled “PCIe Medium-Specific Header” and “PCIe Medium-Specific Trailer” are specific to  
 257 carrying MCTP packets using PCIe VDMs. The fields labeled “MCTP Transport Header” and “MCTP Packet  
 258 Payload” are common fields for all MCTP packets and messages and are specified in [MCTP](#). This document  
 259 defines the location of those fields when they are carried in a PCIe VDM. The PCIe specification allows the  
 260 last four bytes of the PCIe VDM header to be vendor defined. The MCTP over PCIe VDM transport binding  
 261 specification uses these bytes for MCTP Transport header fields under the DMTF Vendor ID. This  
 262 document also specifies the *medium-specific* use of the MCTP “Hdr Version” field.

263 Table 1 lists the PCIe medium-specific fields and field values that shall be used in MCTP over PCIe VDM  
 264 communications. When not specified, field values shall be set according to PCIe specifications. Note that  
 265 the presence of TLP prefixes in MCTP over PCIe VDM packets is implementation dependent and outside  
 266 the scope of this specification.

267

**Table 1 – PCI Express medium-specific MCTP packet fields**

Field	Description
R or Fmt[2]	PCIe 1.1/2.0: PCIe reserved bit (1 bit). PCIe 2.1 and above: Fmt[2]. Set to 0b.
Fmt	Format (2 bits). Set to 11b to indicate 4 dword header with data.
Type	Type and Routing (5 bits). [4:3] Set to 10b to indicate a message [2:0] PCI message routing (r2r1r0) 000b : Route to Root Complex 010b : Route by ID 011b : Broadcast from Root Complex Other routing fields values are not supported for MCTP.

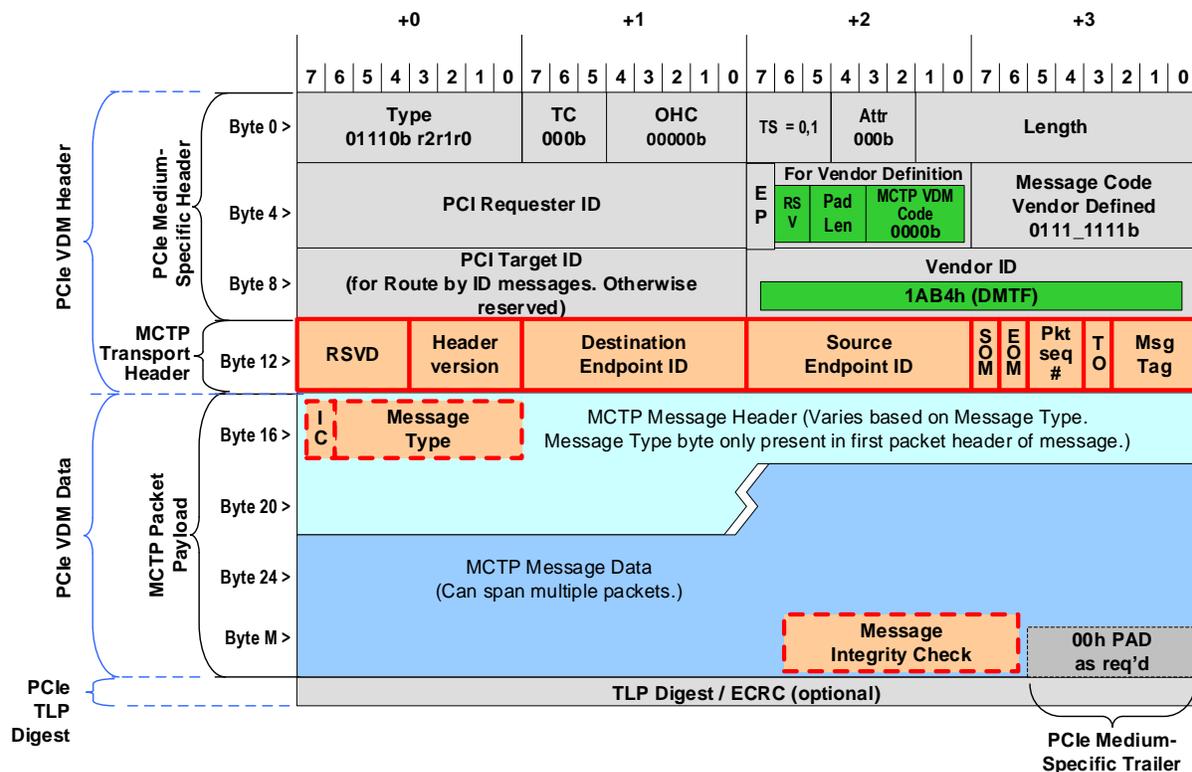
Field	Description
R or T9	PCIe 1.1/2.0/2.1/3.x: PCIe reserved bit (1 bit). Set to 0b. PCIe 4.x and above: T9 (1bit). Set to 0b.
TC	Traffic Class (3 bits). Set to 000b for MCTP over PCIe VDM.
R or R   Attr   R   TH or T8   Attr   LN   TH	PCIe 1.1/2.0: PCIe reserved bits (4 bits). Set to 0000b PCIe 2.1/3.x: PCIe reserved bit (1 bit), Attr[2] (1 bit) – Set to 0b, reserved bit (1bit), and TH (1bit) – Set to 0b. PCIe 4.x and above: T8 bit (1 bit) – Set to 0b, Attr[2] (1 bit) – Set to 0b, LN (1bit) – Set to 0b, and TH (1bit) – Set to 0b
TD	TLP Digest (1 bit). 1b indicates the presence of the TLP Digest field at the end of the PCIe TLP (transaction layer packet). The TD bit should be set in accordance with the devices overall support for the TLP Digest capability, and whether that capability is enabled. See description of the TLP Digest / ECRC field, below, for additional information. Note that earlier versions of this specification erroneously required this bit to be set to 0b, which would have required devices to not support the TLP Digest capability.
EP	Error Poisoned (1 bit).
Attr[1:0]	Attributes (2 bits). Set to 00b or 01b for all MCTP over PCIe VDM.
R or AT	PCIe 1.1: PCIe reserved bits (2 bits). PCIe 2.0 and above: Address Type (AT) field. Set to 00b.
Length	Length: Length of the PCIe VDM Data in dwords. Implementations shall support the baseline transmission unit defined in the <a href="#">MCTP Base Specification</a> . For example, supporting a baseline transmission unit of 64 bytes requires supporting PCIe VDM data up to 16 dwords. An implementation may optionally support dword aligned larger transfer unit sizes.
PCI Requester ID	Bus/device/function or bus/function number of the managed endpoint sending the message.
Pad Len	Pad Length (2-bits). 1-based count (0 to 3) of the number of 0x00 pad bytes that have been added to the end of the packet to make the packet dword aligned with respect to PCIe. Because only packets with the EOM bit set to 1b are allowed to be less than the transfer unit size, packets that have the EOM bit set to 0b will already be dword aligned and will thus not require any pad bytes and will have a pad length of 00b.
MCTP VDM Code	Value that uniquely differentiates MCTP messages from other DMTF VDMs. Set to 0000b for this transport mapping as defined in this specification.
Message Code	(8 bits). Set to 0111_1111b to indicate a Type 1 VDM.
PCI Target ID	(16 bits). For Route by ID messages, this is the bus/device/function number or bus/function number that is the physical address of the target endpoint. This field is ignored for Broadcast and for Route to Root Complex messages.
Vendor ID	(16 bits). Set to <b>6836</b> (0x1AB4) for DMTF VDMs. The most significant byte is in byte 10, the least significant byte is byte 11.
RSVD	MCTP reserved (4 bits). Set these bits to 0 when generating a message. Ignore them on incoming messages.
Hdr Version	MCTP version (4 bits) 0001b : For MCTP devices that conform to the <a href="#">MCTP Base Specification</a> and this version of the PCIe VDM transport binding. All other settings: Reserved to support future packet header field expansion or header version.

Field	Description
00h PAD	Pad bytes. 0 to 3 bytes of 00h as required to fill out the overall PCIe VDM data to be an integral number of dwords. Because only packets with the EOM bit set to 1b are allowed to be less than the transfer unit size, packets that have the EOM bit set to 0b will already be dword aligned and will thus not require any pad bytes and will have a pad length of 00b.
TLP Digest / ECRC	(32 bits). TLP Digest / ECRC (End-to-end CRC). This field is defined for all PCIe TLPs (Transaction Layer Packets). Device support for this field is optional. However, per <a href="#">PCIe v2.1</a> and above: "If a device Function is enabled to generate ECRC, it must calculate and apply ECRC for all TLPs originated by the Function. If the device supports generating this field, it must support it for all TLPs." Additionally, per <a href="#">PCIe v2.1</a> and above, if the ultimate PCI Express Receiver of the TLP does not support ECRC checking, the receiver must ignore the TLP Digest.

268 Note: In this table, "PCIe X.X and above" means up to the latest PCIe version covered by this  
 269 specification.

270 **6.2.2 Flit Mode**

271 Figure 2 and Figure 3 show the encapsulation of MCTP packet fields within a PCIe VDM in Flit Mode with  
 272 or without OHCs.



273  
 274 **Figure 2 – MCTP over PCI Express Vendor Defined Message (VDM) packet format**  
 275 **(Flit Mode within a Segment without IDE)**

276



**Table 2 – PCI Express medium-specific MCTP Packet Fields in Flit Mode**

Field	Description
<b>PCIe Header</b>	
Type	Type and Routing (8 bits). [7:3] Set to 01110b to indicate a message [2:0] PCI message routing (r2r1r0) 000b : Route to Root Complex 010b : Route by ID 011b : Broadcast from Root Complex Other routing fields values are not supported for MCTP.
TC	Traffic Class (3 bits). Set to 000b for MCTP over PCIe VDM.
OHC	Indicates the type of OHCs present after the header. Possible values are: 00000b: no OHCs 00100b: Indicates OHC-C is present 00101b: Indicates OHC-A4 and OHC-C are present
TS	TS[2:0] field indicates Trailer Size. The possible encodings for PCIe VDMs are: 000b – No Trailer 001b – 1 dword Trailer containing ECRC 101b – 3 dwords Trailer with IDE MAC if and only if OHC-C present and indicates IDE TLP 110b – 4 dwords Trailer with IDE MAC and PCRC if and only if OHC-C present and indicates IDE TLP
Attr[2:0]	Attributes (3 bits). Set to 000b for all MCTP over PCIe VDM.
Length	Length: Length of the PCIe VDM Data in dwords. Implementations shall support the baseline transmission unit defined in the <a href="#">MCTP Base Specification</a> . For example, supporting a baseline transmission unit of 64 bytes requires supporting PCIe VDM data up to 16 dwords. An implementation may optionally support larger transfer unit sizes.
PCI Requester ID	Bus/device/function or bus/function number of the managed endpoint sending the message.
Pad Len	Pad Length (2-bits). 1-based count (0 to 3) of the number of 0x00 pad bytes that have been added to the end of the packet to make the packet dword aligned with respect to PCIe. Because only packets with the EOM bit set to 1b are allowed to be less than the transfer unit size, packets that have the EOM bit set to 0b will already be dword aligned and will thus not require any pad bytes and will have a pad length of 00b.
MCTP VDM Code	Value that uniquely differentiates MCTP messages from other DMTF VDMs. Set to 0000b for this transport mapping as defined in this specification.
Message Code	(8 bits). Set to 0111_1111b to indicate a Type 1 VDM.
PCI Target ID	(16 bits). For Route by ID messages, this is the bus/device/function number or bus/function number that is the physical address of the target endpoint. This field is ignored for Broadcast and for Route to Root Complex messages.
Vendor ID	(16 bits). Set to <b>6836</b> (0x1AB4) for DMTF VDMs. The most significant byte is in byte 10, the least significant byte is byte 11.
<b>Orthogonal Headers (OHC)</b>	
OHC-A4	Includes the Destination Segment and Destination Segment Valid Note: The PASID and PSV fields in OHC-A4 are not relevant to MCTP VDMs.
OHC-C	Includes Requester Segment, RSV (Requester Segment Valid) and IDE parameters.
<b>MCTP Header</b>	

Field	Description
RSVD	MCTP reserved (4 bits). Set these bits to 0 when generating a message. Ignore them on incoming messages.
Hdr Version	MCTP version (4 bits) 0001b : For MCTP devices that conform to the <a href="#">MCTP Base Specification</a> and this version of the PCIe VDM transport binding. All other settings: Reserved to support future packet header field expansion or header version.
<b>MCTP Trailer</b>	
00h PAD	Pad bytes. 0 to 3 bytes of 00h as required to fill out the overall PCIe VDM data to be an integral number of dwords. Because only packets with the EOM bit set to 1b are allowed to be less than the transfer unit size, packets that have the EOM bit set to 0b will already be dword aligned, and will thus not require any pad bytes and will have a pad length of 00b.
<b>PCIe Trailer (3 options if exists)</b>	
TLP Digest / ECRC /	(32 bits). TLP Digest / ECRC (End-to-end CRC).
IDE MAC	(96 bits). IDE Message Authentication Code (MAC).
IDE MAC + PCRC	(128 bits). IDE Message Authentication Code (MAC) + Plaintext CRC

292 **6.3 Supported media**

293 This physical transport binding has been designed to work with the following media as defined in [DSP0239](#)  
 294 and listed in Table 3. Use of this binding with other types of physical media is not covered by this  
 295 specification. Refer to DSP0239 for all supported physical media by MCTP transport bindings.

296 An implementation that is compliant with this specification shall at least support one of the PCIe media  
 297 listed in Table 3. Note that the CXL is built on the [PCI Express](#) (PCIe) physical and electrical interface.

298 **Table 3 – Supported media**

Physical Media Identifier	Description
0x08	PCIe revision 1.1 compatible
0x09	PCIe revision 2.0 compatible
0x0A	PCIe revision 2.1 compatible
0x0B	PCIe revision 3.x compatible
0x0C	PCIe revision 4.x compatible
0x0D	PCIe revision 5.x compatible, CXL 1.x/2.x compatible
0x0E	PCIe revision 6.x Non-Flit Mode Compatible,
0x40	PCIe revision 6.x Flit Mode Compatible, CXL 3.x compatible
0x41	PCIe revision 7.x compatible, CXL 4.x compatible

299 Note: The compatibility mentioned above is to a specific PCIe specification version and not to a specific  
 300 speed. For example, a device reporting compatibility to PCIe 4.x may still operate in Gen3 or lower  
 301 speeds.

302 **6.4 Physical address format for MCTP control messages**

303 The address format shown in Table 4 (Non-Flit Mode) and Table 5 (Flit Mode) is used for MCTP control  
 304 commands that require a physical address parameter to be returned for a bus that uses this transport  
 305 binding with one of the supported media types listed in 6.3 This includes commands such as the Resolve  
 306 Endpoint ID, Routing Information Update, and Get Routing Table Entries commands.

307 **Table 4 – Physical address format (Non-Flit Mode)**

Format Size	Address Type	Layout and Description	
2 bytes (BDF ID)	Bus Device Function (BDF)	byte 1	[7:0] – Bus number
		byte 2	[7:3] – Device number [2:0] – Function number
2 bytes (ARI ID)	Alternate Routing Identifier (ARI)	byte 1	[7:0] – Bus number
		byte 2	[7:0] – Function number

308 **Table 5 – Physical address format (Flit Mode)**

Format Size	Address Type	Layout and Description	
3 bytes (Segment, BDF ID)	Segment, Bus Device Function (BDF)	byte 1	[7:0] – Segment number
		byte 2	[7:0] – Bus number
		byte 3	[7:3] – Device number [2:0] – Function number
3 bytes (Segment, ARI ID)	Segment, Alternate Routing Identifier (ARI)	byte 1	[7:0] – Segment number
		byte 2	[7:0] – Bus number
		byte 3	[7:0] – Function number

309 Note: If a message is received in Non-Flit Mode or in Flit Mode without OHC-A4 present, or with OHC-A4  
 310 present but DSV bit cleared, then the segment number used to create the physical address is implicitly  
 311 set to the local segment. In the same manner, to translate a physical address received in Non-Flit Mode  
 312 format to Flit Mode format, the local segment number shall be added.

313 **6.5 Message routing**

314 Physical packet routing within a PCIe bus uses routing as defined by the PCIe specification. PCIe  
 315 physical routing/bridging is not the same as MCTP bridging. PCIe physical routing/bridging is generally  
 316 transparent to MCTP. There are no MCTP-defined functions for configuring or controlling the setup of a  
 317 PCIe bus. The following types of PCIe addressing are used with MCTP messages:

318 **• Route by ID**

319 All MCTP over PCIe VDM packets between endpoints that are not the bus owner shall use  
 320 Route by ID for message routing.

321 The MCTP bus owner shall use Route by ID for messages to individual MCTP endpoints.

322 MCTP endpoints are required to capture the PCIe source physical address (including segment  
 323 ID in Flit Mode) and the MCTP source EID when receiving an EID assignment MCTP control  
 324 request message. This is because this request can only be issued by the MCTP bus owner.

325 Any MCTP PCIe VDM message received with this routing and Destination EID set to 0xFFh  
 326 (Broadcast ID) should be dropped.

327       •   **Route to Root Complex (RC)**

328           MCTP endpoints shall use this routing for the Discovery Notify request message to the MCTP  
329           bus owner as part of the MCTP over PCIe VDM discovery process.

330           The MCTP endpoints shall use this routing for responding to the MCTP control request  
331           messages that were sent using Broadcast from Root Complex.

332           If the MCTP bus owner is in the PCIe root complex, then these messages should be directly  
333           accessed by the MCTP bus owner.

334           In designs where the PCIe MCTP Bus Owner is not in the PCIe root complex, these messages  
335           shall be relayed to the MCTP bus owner. The communication of MCTP PCIe VDM packets that  
336           are destined to the MCTP bus owner outside the PCIe root complex relies on the root complex  
337           relaying any MCTP over PCIe VDM messages received using Route to Root Complex to the  
338           MCTP bus owner. That communication is implementation-specific and is outside the scope of  
339           this specification. A receiver of an MCTP PCIe VDM message with this routing may ignore the  
340           Destination EID.

341       •   **Broadcast from Root Complex (RC)**

342           The MCTP bus owner should use Broadcast from Root Complex (RC) for the transmission of  
343           Prepare for Endpoint Discovery and Endpoint Discovery messages.

344           If the MCTP bus owner is in the PCIe root complex, the MCTP bus owner should use PCIe root  
345           complex for this routing for the Prepare for Endpoint Discovery and Endpoint Discovery  
346           messages as part of the MCTP over PCIe VDM discovery process. If the MCTP bus owner is  
347           not in the Root Complex, then the MCTP bus owner may use an implementation-specific  
348           scheme to signal the transmission of Prepare for Endpoint Discovery and Endpoint Discovery  
349           messages.

350           If any other MCTP PCIe VDM message is received with this routing, then the message should  
351           be dropped. Any MCTP PCIe VDM message using this routing should use a destination EID of  
352           0xFF. A receiver of an MCTP PCIe VDM message with this routing may ignore the Destination  
353           EID.

354

355   **6.5.1 Routing peer transactions on a PCIe bus**

356           The PCIe specification does not require peer-to-peer routing support in PCIe root complexes or switches.  
357           For this reason, MCTP over PCIe VDM messages may need to be routed through an MCTP bridge in the  
358           MCTP bus owner. When peer-to-peer routing of MCTP PCIe VDM messages between two MCTP  
359           endpoints requires routing through a PCIe root complex and the PCIe root complex does not support  
360           peer-to-peer routing, then all MCTP over PCIe VDM messages between two MCTP endpoints shall be  
361           routed to or through an MCTP bridge located in the PCIe root complex. If the MCTP bus owner is in the  
362           PCIe root complex, and the PCIe root complex supports peer-to-peer routing, then the PCIe root complex  
363           shall use direct physical addressing to support routing between two MCTP endpoints on the PCIe bus for  
364           the non-flit mode.

365   **6.5.2 Routing messages between PCIe and other buses**

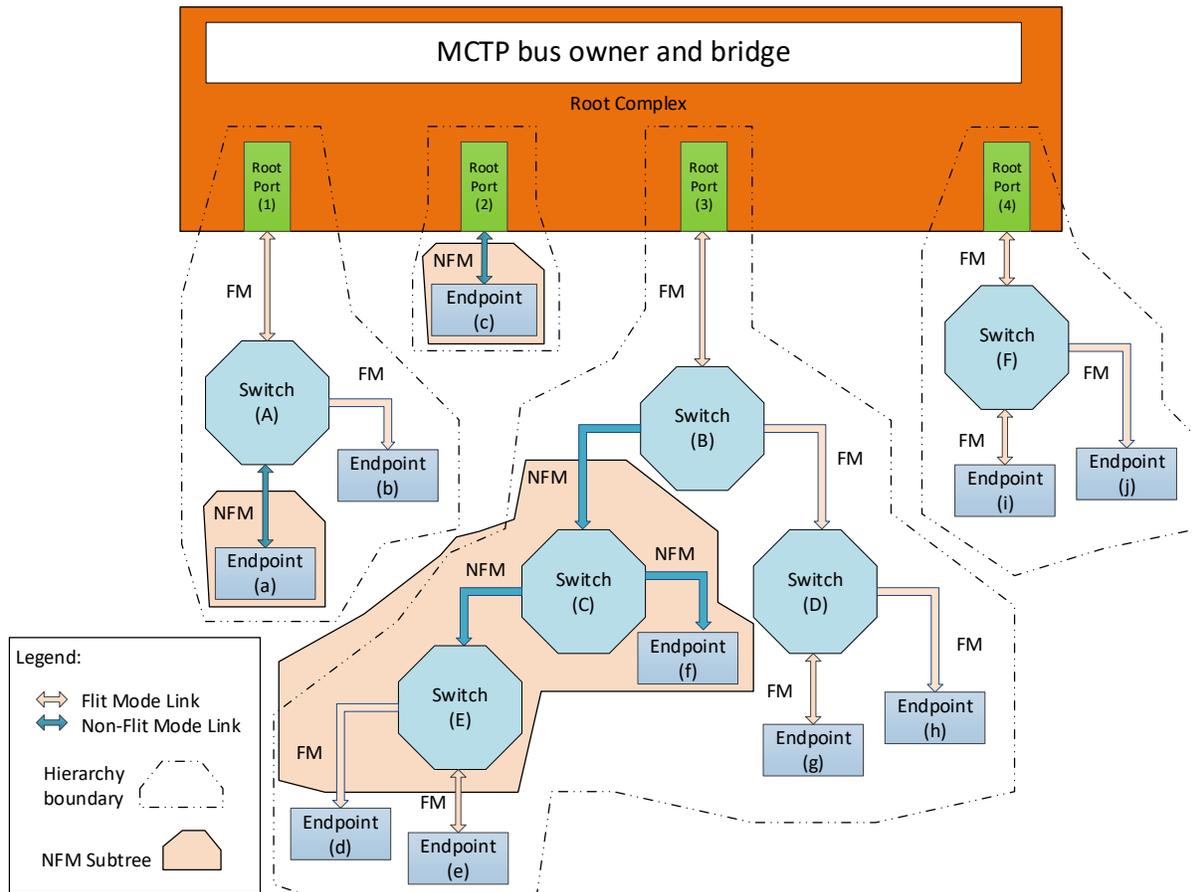
366           All MCTP messages that span between PCIe and other buses shall be sent through the MCTP bus  
367           owner. The MCTP bus owner has the destination EID routing tables necessary to route messages  
368           between the two bus segments.

369           If an endpoint is aware of multiple routes to a destination over multiple bus types, a higher-level  
370           algorithm/protocol above MCTP shall be used to determine which bus/route to use. Typically, this  
371           decision can be based on facts such as power state and MCTP discovery state.

372 **6.5.3 Routing Messages in Between Flit Mode and Non-Flit Mode domains**

373 Generally, PCIe rules for routing between segments and presence and values of the Requester Segment  
 374 and Destination Segment need to be followed so that MCTP can route "transparently" in the presence of  
 375 PCIe Flit Mode domains.

376 Figure 4 describes the possible transitions between Flit Mode and Non-Flit Mode domains:



377  
 378 **Figure 4 – Flit Mode to Non-Flit Mode routing options**

379 **Table 6 – Address Translations when transitioning between Flit Mode and Non-Flit Mode**  
 380 **hierarchies**

Scenario	Fields translation by Root Port/Bridge	Requires MCTP Bridge?	Example (based on Figure 4)
Flit Mode to Non-Flit Mode or Flit Mode to Flit Mode within segment	Use Target ID and Requester ID as is.	No	(b) to (a) or (i) to (j)

Scenario	Fields translation by Root Port/Bridge	Requires MCTP Bridge?	Example (based on Figure 4)
Flit Mode to Non-Flit Mode across segments	Route to segment indicated by EID lookup.  Requester ID in destination segment = {bridge BDF}  MCTP bridge shall replace the target ID based on EID lookup.	Yes	(i) to (f)
Flit Mode to Flit Mode across segments	Route to segment indicated by Destination Segment. Target ID kept as is. Requester ID = source Requester ID (no change by MCTP bridge)	No	(j) to (g)
Non-Flit Mode to Flit Mode within segment	Use Target ID and Requester ID as is.	No	(f) to (g)
Non-Flit Mode to Flit Mode across segments	Route to segment indicated by EID lookup.  Requester ID in destination segment = {source segment, source BDF}  MCTP bridge shall replace the target ID based on EID lookup.	Yes	(f) to (i)
Non-Flit Mode through Flit Mode fabric across segments	Route to segment indicated by EID lookup.  Requester ID in destination segment = Bridge ID  MCTP bridge shall replace the Target ID based on EID lookup. The MCTP bridge(s) on which Non-Flit Mode island resides shall be aware of it and translate accordingly.	Yes	(e) to (c) or (e) to (i)
Flit Mode to Flit Mode in a Non-Flit Mode island	Use Target ID and Requester ID as is. Segment ID is ignored for the routing.  Note: The segment ID seen by these endpoints may be different from the actual ID of the segment as described in the <a href="#">PCIe base specification</a> .	No	(e) to (d)

381

382 When a message crosses the boundary between Flit Mode and Non-Flit Mode hierarchies, the Segment  
383 part of the address may be removed (Flit Mode to Non-Flit Mode) or added (Non-Flit Mode to Flit Mode).  
384 Table 6 describes the translations done when a message crosses between Flit Mode and Non-Flit Mode  
385 hierarchies in various scenarios. The “Requires MCTP Bridge” column in Table 6 indicates whether the  
386 described scenario requires one or more PCIe Root Ports to function as an MCTP bridge, as [DSP0236](#)  
387 describes. The “Example” column in Table 6 provides an example of the described scenario using the  
388 topology shown in Figure 4.

389 To identify the physical address to use, an endpoint may use the Resolve Endpoint ID command that  
 390 translates an EID to a physical address. As endpoints in Non-Flit Mode may not be aware of the Flit Mode  
 391 address format, when such an endpoint is requesting the address of a Flit Mode device it should receive  
 392 a physical address it knows how to use, hence the address shall not include a segment number. In case  
 393 the resolved EID is in a different segment, the physical address shall point to an MCTP bridge that can  
 394 translate and route based on the destination EID to a physical address in another segment.

395 Note: This mechanism assumes the EIDs are unique across all segments, and that physical addressing is  
 396 not used when crossing between Flit Mode and Non-Flit Mode hierarchies.

397 A bus owner may discover the Physical medium supported by an endpoint using the Query Supported  
 398 Interfaces command defined in [DSP0236](#). If the command is not supported, and the bus owner has no  
 399 other method to detect the supported medium, then a Non-Flit Mode support shall be assumed.

400 **6.5.4 Example of Resolve Endpoint ID response**

401 The table below describes possible responses to Resolve Endpoint ID requests based on the requester  
 402 and the requested target. The examples are based on Figure 4:

403 **Table 7 – Address used for routing examples**

Requester	Requester target	Responding MCTP bus owner	Response (Physical address)
b	a	Root Port 1	{B,D,F} of (a)
i	j	Root Port 4	{S,B,D,F} of (j)
i	f	Root Port 4	{S,B,D,F} of MCTP bridge 4
j	g	Root port 4	{S,B,D,F} of (g)
f	g	Root port 3	{B,D,F} of MCTP bridge 3
f	i	Root port 3	{B,D,F} of MCTP bridge 3
e	c	Root port 3	{B,D,F} of MCTP bridge 3
e	i	Root port 3	{B,D,F} of MCTP bridge 3
e	f	Root port 3	{B,D,F} of (f)

404 Notes:

- 405 • The MCTP bus owner may choose to return its own address instead of the endpoint address in all  
 406 cases. In this case, all the traffic will be bridged. The chosen routing method is implementation  
 407 dependent and is outside of the scope of this specification.
- 408 • When a {B,D,F} of MCTP bridge is mentioned, it can be any function within the root complex acting  
 409 as the MCTP bus owner.

410 **6.6 Bus owner address**

411 The MCTP PCIe VDM bus owner functionality shall be accessible through “Route to Root Complex”  
 412 addressing.

413 **6.7 Bus and Segment address assignment for PCIe**

414 PCIe bus and segment addresses are assigned by the mechanisms specified in [PCIe](#).

## 415 6.8 Host dependencies

416 MCTP over PCIe VDM, when used in a typical “PC” computer system, has a dependency on the host  
417 CPU, host software, power management states, link states, and reset. Some of these dependencies are  
418 described as follows:

- 419 • **Reset**

420 Assertion of “Fundamental Reset” or “Conventional Reset” on the bus causes both the host  
421 functionality as well as the MCTP PCIe VDM communication on an MCTP endpoint to be reset.  
422 From the assertion “Fundamental Reset” or “Conventional Reset” until a physical address is  
423 assigned as part of the PCIe fabric enumeration, no MCTP over PCIe VDM messages can be  
424 sent to the MCTP endpoint.

425 Similarly, if MCTP PCIe VDM communication is supported on a function of a PCIe device, a  
426 function level reset (FLR) may reset MCTP PCIe VDM endpoint as well as MCTP PCIe VDM  
427 communication on that function. An MCTP PCIe VDM endpoint may not be able to respond to  
428 PCIe VDMs during FLR. The PCIe device may use mechanisms outside of this specification to  
429 notify function’s FLR to delay any PCIe VDM communication until the FLR processing is  
430 complete.

431 A PCIe hot reset may reset a MCTP PCIe VDM endpoint. An implementation should retain the  
432 EID during a PCIe hot reset.

433 An MCTP PCIe VDM endpoint that is reset may need to be reinitialized and/or rediscovered.

- 434 • **Configuration and enumeration**

435 Following the de-assertion “Fundamental Reset” or “Conventional Reset”, the software running  
436 on the host CPU configures and enumerates the PCIe fabric. Failure of the host CPU or boot  
437 software to properly configure and enumerate the PCIe fabric prevents it from being used for  
438 MCTP over PCIe VDM messaging.

- 439 • **Power management states**

440 The host (as defined in the context of the [PCI Express specification](#)) controls PCIe bus power  
441 management. The host may power down PCIe devices and links, or place them in sleep states,  
442 independent of management controllers, which may cause MCTP PCIe VDM communication to  
443 be unavailable. An MCTP PCIe VDM endpoint on a PCIe device may not be able to respond to  
444 PCIe VDMs in low power states or sleep states. The PCIe device may notify using mechanisms  
445 outside of this specification of the current power state to delay any PCIe VDM communication  
446 until the PCIe device transitions to a state where the PCIe VDM communication is available.  
447 Depending on the device usage in the system, a PCIe device may retain or lose states such as  
448 EID, “discovered” state, and routing information (if the device is an MCTP bridge). A PCIe  
449 device that loses MCTP PCIe VDM communication state needs to be reinitialized and/or  
450 rediscovered after it returns to a power state that supports MCTP over PCIe VDM  
451 communication.

- 452 • **Link states**

453 The PCIe link states affect MCTP over PCIe VDM communications. MCTP over PCIe VDM  
454 communication can be performed only when the PCIe link is in a state that allows VDM  
455 communications. The mechanisms for PCIe link state transitions are outside the scope of this  
456 specification.

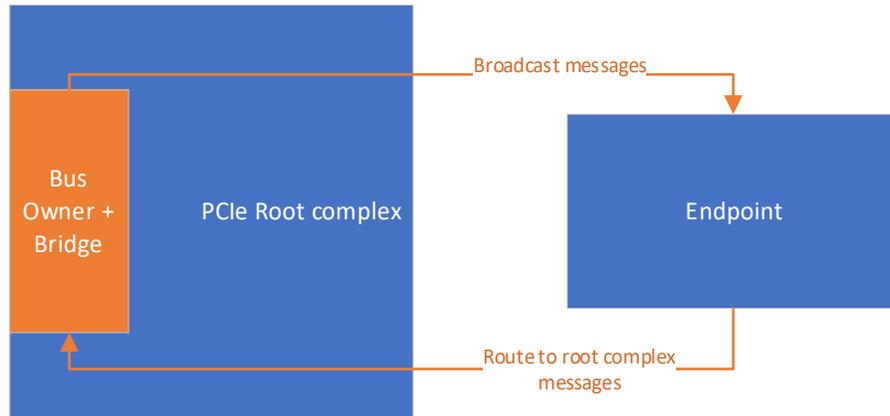
- 457 • **PCIe Root Complex**

458 The PCIe Root Complex (RC) is responsible for communicating Route to Root Complex MCTP  
459 over PCIe VDM discovery messages to the MCTP bus owner.

460 As stated above, the MCTP bus owner may be co-located with the Root Complex or may be  
461 located in one of the endpoints (for example, the MC). In the second case, the Root Complex

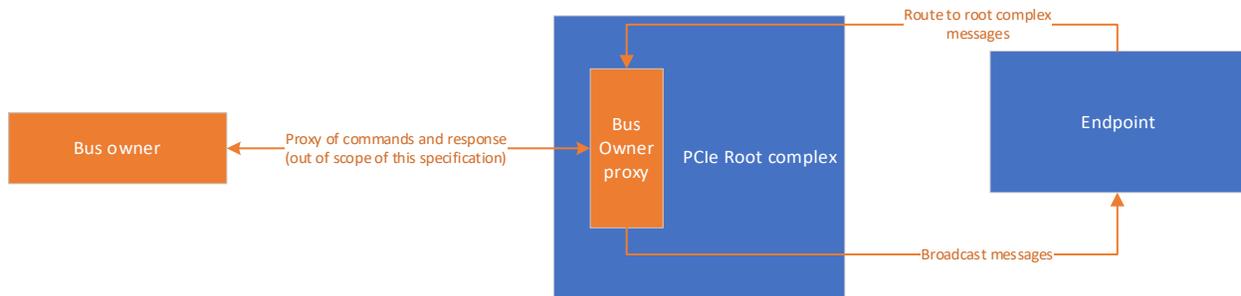
462 needs to provide a method for the MCTP bus owner to receive Route to Root Complex  
 463 messages and transmit messages via the Root Complex, including broadcast, Route to Root  
 464 Complex, and unicast messages.

465 Figure 5 and Figure 6 describe these two options for the location of the bus owner.



466

467 **Figure 5 – Bus owner and Root Complex co-located**



468

469 **Figure 6 – Bus owner and Root Complex separated**

470 **6.9 Discovery Notify message use for PCIe**

471 An MCTP control Discovery Notify message shall be sent from a PCIe endpoint to the MCTP bus owner  
 472 whenever the physical address for the device changes (that is, the endpoint receives a Type 0  
 473 configuration write request and the bus number and/or the segment number is different from the currently  
 474 stored bus and segment numbers) and the assigned bus number and optional segment number have  
 475 been captured by the endpoint. This occurs on the first Type 0 configuration write following a PCIe bus  
 476 reset during initial enumeration, or during re-enumeration where the bus or segment number has  
 477 changed (for example, because of a hot plug event, bus reset, and so on).

478 Endpoints use the Discovery Notify command to inform the MCTP bus owner that it needs to update the  
 479 endpoint's ID. The Discovery Notify command shall be sent with the PCIe message routing set to 000b

480 (Route to Root Complex), the Destination Endpoint ID for the Discovery Notify message shall be set to  
481 the Null Destination EID. The Source Endpoint ID field shall be set to the Null Source EID if the device  
482 has not yet been assigned an EID; otherwise, it shall contain the assigned EID value. If the MCTP bus  
483 owner is not in the PCIe root complex, the PCIe root complex provides Discovery Notify command  
484 information to the MCTP bus owner using implementation specific mechanisms. These mechanisms are  
485 outside the scope of this specification.

## 486 **6.10 MCTP over PCIe endpoint discovery**

487 This clause describes the steps used to support discovering MCTP endpoints on PCIe.

### 488 **6.10.1 Discovered flag**

489 Each endpoint (except the bus owner) on the PCIe bus maintains an internal flag called the *Discovered*  
490 flag.

491 The flag is set to the *discovered* state when the Set Endpoint ID command is received.

492 The Prepare for Endpoint Discovery message causes each recipient endpoint on the PCIe bus to set their  
493 respective Discovered flag to the *undiscovered* state. For the Prepare for Endpoint Discovery request  
494 message, the routing in the physical transport header should be set to `011b` (Broadcast from Root  
495 Complex). An endpoint also sets the flag to the *undiscovered* state at the following times:

- 496 • Whenever the PCIe physical address associated with the endpoint is initially assigned or is  
497 changed to a different value.
- 498 • Whenever an endpoint first appears on the bus and requires an EID assignment. A device shall  
499 have been enumerated on PCI and have a bus/device/function or bus/function number before it  
500 can do this.
- 501 • During operation, if an endpoint enters a state that causes it to lose its EID assignment.
- 502 • For endpoints that have already received an EID assignment but are in any temporary state  
503 where the endpoint was unable to respond to MCTP control requests for more than  $T_{RECLAIM}$   
504 seconds.

505 Only endpoints that have their Discovered flag set to *undiscovered* shall respond to the Endpoint  
506 Discovery message. Endpoints that have the flag set to *discovered* shall not respond to the Endpoint  
507 Discovery message.

508 For PCIe endpoints, an Endpoint Discovery broadcast request message can be sent by the MCTP bus  
509 owner to discover all MCTP-capable devices. MCTP-capable endpoints respond with an Endpoint  
510 Discovery response message.

511 An MCTP-capable endpoint shall respond to broadcast MCTP control request messages only if a PCI bus  
512 number and in Flit-Mode, a segment number and a bus number are assigned to the associated PCIe  
513 function; otherwise, the endpoint should silently discard such MCTP messages.

### 514 **6.10.2 PCIe endpoint announcement**

515 One or more endpoints may announce their presence and their need for an EID assignment by  
516 autonomously sending a Discovery Notify message to the bus owner. This would typically trigger the  
517 MCTP bus owner to perform the PCIe endpoint discovery/enumeration processes described in the  
518 following subclauses.

### 519 6.10.3 Full endpoint Discovery/Enumeration

520 The following process is typically used when the MCTP bus owner wishes to discover and enumerate all  
521 MCTP endpoints on the PCIe bus.

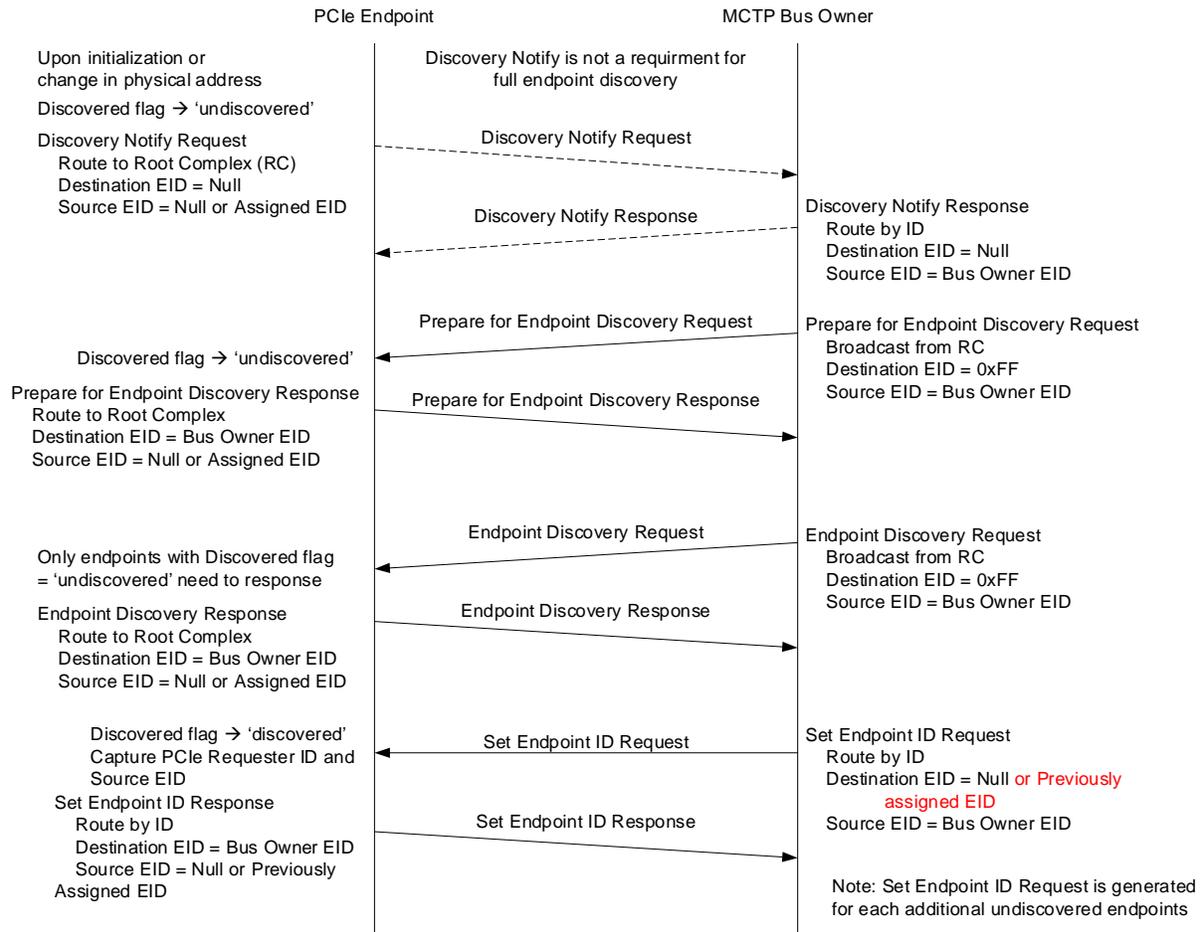
- 522 1) The MCTP bus owner issues a broadcast Prepare for Endpoint Discovery message. This  
523 message causes each discoverable endpoint on the bus to set its PCIe endpoint Discovered  
524 flag to undiscovered. Depending on the number of endpoints and the buffer space available in  
525 the MCTP bus owner, the MCTP bus owner may not receive all of the response messages. The  
526 discovery process does not require the MCTP bus owner to receive all the response messages  
527 to the Prepare for Endpoint Discovery request. Because the MCTP bus owner cannot determine  
528 that all endpoints have received the Prepare for Endpoint Discovery request, it is recommended  
529 that Prepare for Endpoint Discovery request is retried MN1 times to help ensure that all  
530 endpoints have received the request. The MCTP bus owner is not required to wait for MT2 time  
531 interval between the retries.
- 532 2) The MCTP bus owner should wait for MT2 time interval to help ensure that all endpoints that  
533 received the Prepare for Endpoint Discovery request have processed the request.
- 534 3) The MCTP bus owner issues a broadcast Endpoint Discovery request message. All MCTP-  
535 capable devices that have their Discovered flag set to undiscovered will respond with an  
536 Endpoint Discovery response message.
- 537 4) Depending on the number of endpoints and the buffer space available in the MCTP bus owner,  
538 the MCTP bus owner receives some or all these response messages. For each response  
539 message received from an undiscovered MCTP-capable device PCIe bus/device/function or  
540 bus/function number, the MCTP bus owner issues a Set Endpoint ID command to the physical  
541 address for the endpoint. This causes the endpoint to set its Discovered flag to *discovered*.  
542 From this point, the endpoint shall not respond to the Endpoint Discovery command until  
543 another Prepare for Endpoint Discovery command is received, or some other condition causes  
544 the Discovered flag to be set back to *undiscovered*.
- 545 5) If the MCTP bus owner received any responses to the Endpoint Discovery request issued in  
546 Step 3, then it shall repeat steps 3 and 4 until it no longer gets any responses to the Endpoint  
547 Discovery request. In this case, then the MCTP bus owner is allowed to send the next Endpoint  
548 Discovery request without waiting for MT2 time interval. If no responses were received by the  
549 MCTP bus owner to the Endpoint Discovery request within the MT2 time interval, then the  
550 discovery process is completed.

551 After the initial endpoint enumeration, it is recommended that the MCTP bus owner maintains a list of the  
552 unique IDs for the endpoints it has discovered and reassigns the same IDs to those endpoints if a  
553 physical address changes during system operation.

554 An MCTP-capable endpoint may respond to Route by ID Prepare for Endpoint Discovery and Endpoint  
555 Discovery request messages.

556 Figure 7 provides an example of flow of operations for full endpoint discovery when the MCTP Bus Owner  
557 resides in the PCIe RC.

**Full PCIe MCTP Endpoint Discovery Example**



558

559

**Figure 7 – Example Flow of operations for full MCTP Discovery over PCIe**

560 Note: In this example flow, the destination EID in Set Endpoint ID Request is set to Null if not assigned  
561 before.

562 **6.10.4 Partial endpoint Discovery/Enumeration**

563 This process is used when the MCTP bus owner wishes to discover endpoints that may have been added  
564 to the bus after full enumeration has been done. This situation can occur if a device has its physical  
565 address change after the full enumeration has been done, or when a hot-plug device is added to the  
566 system, or if a device that is already present in the system—but was in a disabled or powered-down  
567 state—comes on-line.

568 The partial discovery process is the same as the full discovery process except that the MCTP bus owner  
569 skips the step of broadcasting a Prepare for Endpoint Discovery command to avoid clearing the  
570 Discovered flags of already discovered endpoints.

571 The partial discovery process may be initiated when a device that is added or enabled for MCTP sends a  
572 Discovery Notify message to the MCTP bus owner. The MCTP bus owner may also elect to periodically  
573 issue a broadcast Endpoint Discovery message to test whether any undiscovered endpoints have been  
574 missed. The Discovery Notify message provides the MCTP bus owner with the physical address of the

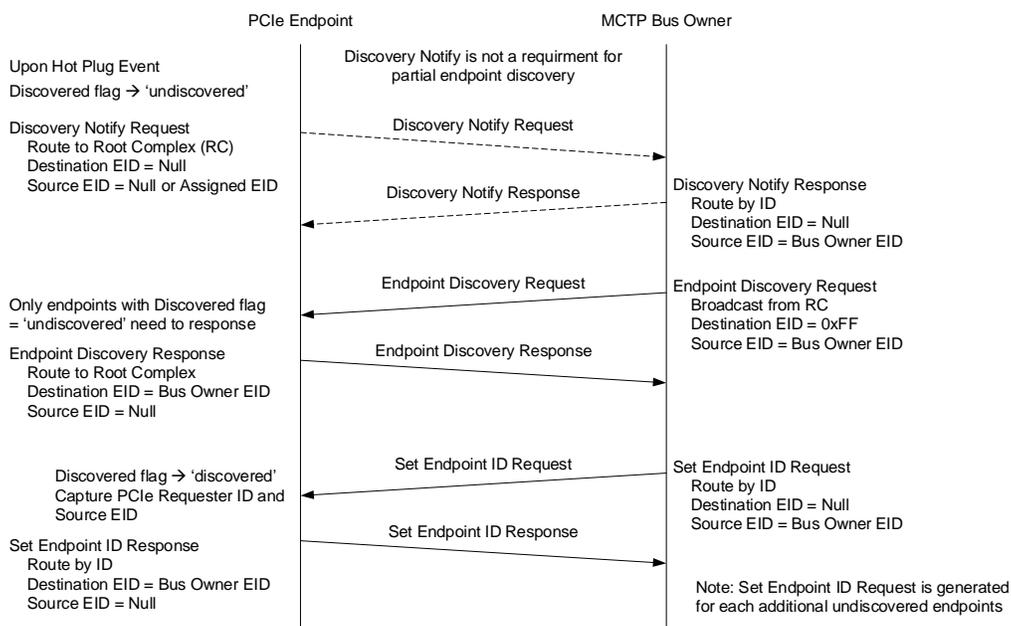
575 MCTP PCIe endpoint. The MCTP bus owner can then send a direct Endpoint Discovery message to the  
 576 endpoint to confirm that the device has not been discovered. The MCTP bus owner then issues a Set  
 577 Endpoint ID command to the physical address for the endpoint which causes the endpoint to set its  
 578 Discovered flag to *discovered*.

579 It is recommended that the MCTP bus owner maintains a list of the unique MCTP EIDs for the endpoints  
 580 it has discovered and reassigns the same MCTP EIDs to those endpoints if a physical address changes  
 581 during system operation.

582 An MCTP-capable endpoint may respond to Route by ID Endpoint Discovery request messages.

583 Figure 8 provides an example of flow of operations for partial endpoint discovery when the MCTP Bus  
 584 Owner resides in the PCIe RC.

Partial PCIe MCTP Endpoint Discovery



585

586

Figure 8 – Flow of operations for Partial Endpoint Discovery

587 **6.10.5 Endpoint re-enumeration**

588 If the bus implementation includes hot-plug devices, the bus owner shall perform a full or partial endpoint  
 589 discovery any time the MCTP bus owner goes into a temporary state where the MCTP bus owner can  
 590 miss receiving a Discovery Notify message (for example, if the bus owner device is reset or receives a  
 591 firmware update). Whether a full or partial endpoint discovery is required is dependent on how much  
 592 information the MCTP bus owner retains from prior enumerations.

593 **6.10.6 EID allocation to PCIe VDM MCTP bus owner**

594 Before EID allocation can start, the bus owner shall have a pool of EIDs to allocate. Since the MCTP bus  
 595 owner of the PCIe bus may not be the topmost MCTP bus owner, this pool should not be pre-allocated. In  
 596 such cases, the PCIe bus owner should provide a method of receiving such an allocation either through a  
 597 proprietary means or by exposing an MCTP bridge that is reachable by the topmost bus owner. With the  
 598 second method, the PCIe bus owner can be allocated a pool using the Allocate Endpoint IDs command.

599 **6.11 MCTP messages timing requirements**

600 Table 8 lists MCTP-specific timing requirements for MCTP messages and operation on the PCIe VDM  
 601 medium. All MCTP Messages over PCIe VDM shall comply with the timing specification listed in Table 8  
 602 unless overridden by the specific message type binding.

603 **Table 8 – Timing specifications for MCTP messages on PCIe VDM**

Timing Specification	Symbol	Min	Max	Description
Endpoint ID reclaim	TRECLAIM	5 sec	—	Maximum interval that an endpoint is allowed to be non-responsive to MCTP control messages before its EID may be reclaimed by the bus owner.  A bus owner shall wait at least for this interval before an EID of the non-responsive endpoint is reclaimed.
Number of request retries	MN1	2	See Description column	Total of three tries, minimum: the original try plus two retries. The maximum number of retries for a given request is limited by the requirement that all retries shall occur within MT4, max of the initial request.
Request-to-response time	MT1	—	120 ms	This interval is measured at the responder from the end of the reception of an MCTP control request to the beginning of the transmission of the corresponding MCTP control response. This requirement is tested under the condition where the responder can successfully transmit the response on the first try.
Time-out waiting for a response	MT2	MT1 max + 2 * MT3 max <sup>[1]</sup>	MT4, min <sup>[1]</sup>	This interval at the requester sets the minimum amount of time that a requester should wait before retrying an MCTP control request. This interval is measured at the requester from the end of the successful transmission of the MCTP control request to the beginning of the reception of the corresponding MCTP control response.  NOTE: This specification does not preclude an implementation from adjusting the minimum time-out waiting for a response to a smaller number than MT2 based on the measured response times from responders. The mechanism for doing so is outside the scope of this specification.
Transmission Delay	MT3	—	3 ms	Allowed time between the end of the transmission of an MCTP Control Protocol message at the transmitter to the beginning of the reception of the MCTP Control Protocol message at the receiver.

Timing Specification	Symbol	Min	Max	Description
Inter-Packet delay for Multi-Packet messages	MT3a	—	100 ms	Allowed time between the end of the transmission of an MCTP packet with EOM=0 to the beginning of the following MCTP packet of the same Message (see the “Message assembly” section of DSP0236), measured at the transmitter
Instance ID expiration interval	MT4	5 sec <sup>[2]</sup>	6 sec	Interval after which the instance ID for a given response will expire and become reusable if a response has not been received for the request. This is also the maximum time that a responder tracks an instance ID for a given request from a given requester.
<p>NOTE 1: Unless otherwise specified, this timing applies to the mandatory and optional MCTP commands.</p> <p>NOTE 2: If a requester is reset, it may produce the same sequence number for a request as one that was previously issued. To guard against this, it is recommended that sequence number expiration be implemented. Any request from a given requester that is received more than MT4 seconds after a previous, matching request should be treated as a new request, not a retry.</p>				

604  
605  
606  
607

## ANNEX A (informative)

### Notations and conventions

#### 608 Notations

609 Examples of notations used in this document are as follows: list into text needed

- 610 • 2:N In field descriptions, this will typically be used to represent a range of byte offsets  
611 starting from byte two and continuing to and including byte N. The lowest offset is on  
612 the left, the highest is on the right.
- 613 • (6) Parentheses around a single number can be used in message field descriptions to  
614 indicate a byte field that may be present or absent.
- 615 • (3:6) Parentheses around a field consisting of a range of bytes indicates the entire range  
616 may be present or absent. The lowest offset is on the left, the highest is on the right.
- 617 • [DSP0236](#) Underlined, blue text is typically used to indicate a reference to a document or  
618 specification called out in Clause 2, "Normative References" or to items hyperlinked  
619 within the document.
- 620 • rsvd Abbreviation for Reserved. Case insensitive.
- 621 • [4] Square brackets around a number are typically used to indicate a bit offset. Bit  
622 offsets are given as 0-based values (that is, the least significant bit [LSb] offset = 0).
- 623 • [7:5] A range of bit offsets. The most significant bit is on the left, the least significant bit is  
624 on the right.
- 625 • 1b The lower case "b" following a number consisting of 0s and 1s is used to indicate the  
626 number is being given in binary format.
- 627 • 0x12A A leading "0x" is used to indicate a number given in hexadecimal format.

628  
629  
630  
631

## ANNEX B (informative)

### Change log

Version	Date	Description
1.0.0	2009-07-28	Initial release
1.0.1	2009-10-30	Created erratum to clarify Length field definition of PCIe VDM header for MCTP PCIe VDM transport binding, modify introduction section, and clean up references section.
1.0.2	2014-12-07	Clarifications to TD bit usage. Added TLP Digest/ECRC to packet figure and to field descriptions table.
1.1.0	2018-10-24	Added support for PCIe Gen 3, PCIe Gen 4, and ARI. Fixed Figure 1 to cover PCIe 1.0/2.0/2.1/3.x/4.0. Clarified MCTP over PCIe VDM compliant management device requirements. Clarified Endpoint ID reclaim definition. Clarified MCTP bus owner requirements in the specification. Eliminated PCIe bus owner term and replaced it with PCIe root complex where applicable.
1.2.0	2021-03-02	Added support for PCIe Gen 5.x, CXL 1.x, and CXL 2.x.
1.2.1	2023-12-12	Added a clarification for bus number assignment before responding to broadcast MCTP control messages. Added clarifications that an MCTP-capable endpoint may respond to Route by ID discovery request messages.
1.3.0	2024-08-09	Added PCIe 6.x Flit Mode and Non-Flit mode support. Fixed full endpoint discovery example flow Addressed the following issues: <ul style="list-style-type: none"> <li>• Removed duplicate reference to DSP0236</li> <li>• Added default action for non-supported parameters</li> <li>• Clarified the distinction between PCIe gen and PCIe rev.</li> <li>• Clarified the destination EID to use per routing method</li> <li>• Clarified behavior under reset or power down conditions.</li> <li>• Allow route by ID for Endpoint Discovery and Endpoint Discovery request messages.</li> <li>• Generalized timing requirements to all message types.</li> </ul>
1.3.1	2025-07-07	Fixed issue #1476 of routing Flit Mode to Non-Flit mode asymmetric routing vs. routing of Non-Flit Mode to Flit mode Fixed issue #1377 When is it safe to send MCTP Discovery Notify over PCIe VDM? Fixed issue #1338 questions on feasibility of an MC as the MCTP PCIe bus owner Fixed issue #1196 Need clarification on EID operation following PCIe Conventional reset
1.4.0	2026-01-16	Added support for PCIe 7 and CXL 4.0 Updated timing table to match other binding specs Added description of bus owner possible locations

		Added description of flow for PCIe bus owner not being topmost bus owner
--	--	--

632