



1  
2  
3  
4

**Document Identifier: DSP0238**

**Date: 2024-08-09**

**Version: 1.3.0**

5 **Management Component Transport Protocol**  
6 **(MCTP) PCIe VDM Transport Binding**  
7 **Specification**

8 **Supersedes: 1.2.1**

9 **Document Class: Normative**

10 **Document Status: Published**

11 **Document Language: en-US**

12 Copyright Notice

13 Copyright © 2009, 2014, 2018, 2021, 2023–2024 DMTF. All rights reserved.

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

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28 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

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

34 PCI-SIG, PCIe, and the PCI HOT PLUG design mark are registered trademarks or service marks of PCI-  
35 SIG. All other marks and brands are the property of their respective owners.

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

# CONTENTS

38	Foreword .....	5
39	Introduction.....	6
40	1 Scope .....	7
41	2 Normative references .....	7
42	3 Terms and definitions .....	8
43	3.1 Destination Physical Address .....	8
44	3.2 Dword.....	8
45	3.3 Flit Mode .....	8
46	3.4 MCTP PCIe Endpoint.....	8
47	3.5 PCIe Segment.....	8
48	3.6 Target Physical Address .....	8
49	4 Symbols and abbreviated terms.....	9
50	4.1 PCIe® .....	9
51	4.2 VDM .....	9
52	4.3 CXL™ .....	9
53	4.4 FM .....	9
54	4.5 NFM .....	9
55	4.6 IDE .....	9
56	4.7 OHC .....	9
57	4.8 TLP.....	9
58	5 Conventions .....	10
59	5.1 Reserved and unassigned values.....	10
60	5.2 Byte ordering.....	10
61	6 MCTP over PCI Express VDM transport.....	10
62	6.1 Overview .....	10
63	6.2 Packet formats .....	10
64	6.3 Supported media.....	16
65	6.4 Physical address format for MCTP control messages.....	17
66	6.5 Message routing .....	17
67	6.6 Bus owner address .....	21
68	6.7 Bus and Segment address assignment for PCIe.....	22
69	6.8 Host dependencies .....	22
70	6.9 Discovery Notify message use for PCIe .....	23
71	6.10 MCTP over PCIe endpoint discovery.....	23
72	6.11 MCTP messages timing requirements.....	26
73	ANNEX A (informative) Notations and conventions.....	28
74	ANNEX B (informative) Change log.....	29
75		

## 76 **Figures**

77	Figure 1 – MCTP over PCI Express Vendor Defined Message (VDM) packet format (Non-Flit Mode) .....	11
78	Figure 2 – MCTP over PCI Express Vendor Defined Message (VDM) Packet Format (Flit Mode within a	
79	Segment without IDE) .....	13
80	Figure 3 – MCTP over PCI Express Vendor Defined Message (VDM) Packet Format (Flit Mode across	
81	segments and/or with IDE).....	14
82	Figure 4 – Flit Mode to Non-Flit Mode routing options.....	19
83	Figure 5 – Example Flow of operations for full MCTP Discovery over PCIe .....	25
84	Figure 6 – Flow of operations for Partial Endpoint Discovery.....	26
85		

## 86 **Tables**

87	Table 1 – PCI Express medium-specific MCTP packet fields.....	11
88	Table 2 – PCI Express Medium-Specific MCTP Packet Fields in Flit Mode.....	15
89	Table 3 – Supported media.....	16
90	Table 4 – Physical address format (Non-Flit Mode) .....	17
91	Table 5 – Physical Address Format (Flit Mode).....	17
92	Table 6 – Address Translations when transitioning between Flit Mode and Non-Flit Mode hierarchies ....	20
93	Table 7 – Address used for routing examples .....	21
94	Table 8 – Timing specifications for MCTP messages on PCIe VDM .....	27
95		

96

## Foreword

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

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

### 101 **Acknowledgments**

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

#### 103 **Editors:**

- 104 • Hemal Shah – Broadcom Inc.
- 105 • Tom Slaight – Intel Corporation
- 106 • Eliel Louzoun – Intel Corporation

#### 107 **Contributors:**

- 108 • Austin Bolen – Dell Technologies
- 109 • Patrick Caporale – Lenovo
- 110 • Steve Glaser – NVIDIA Corporation
- 111 • Brett Henning – Broadcom Inc.
- 112 • Yuval Itkin – NVIDIA Corporation
- 113 • Janusz Jurski – Intel Corporation
- 114 • Mariusz Oriol – Intel Corporation
- 115 • Patrick Schoeller – Hewlett Packard Enterprise
- 116 • Bob Stevens – Dell Technologies

117

## Introduction

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

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

# 125 Management Component Transport Protocol (MCTP) PCIe 126 VDM Transport Binding Specification

## 127 1 Scope

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

## 130 2 Normative references

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

134 CXL Consortium, *Compute Express Link™ (CXL™) Specification Revision 1.0*,  
135 <https://www.computeexpresslink.org>

136 CXL Consortium, *Compute Express Link™ (CXL™) Specification Revision 1.1*,  
137 <https://www.computeexpresslink.org>

138 CXL Consortium, *Compute Express Link™ (CXL™) Specification Revision 2.0*,  
139 <https://www.computeexpresslink.org>

140 DMTF DSP0236, *Management Component Transport Protocol (MCTP) Base Specification 1.3*  
141 [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)

142 DMTF DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes 1.9*  
143 [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)

144 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,  
145 <https://www.iso.org/sites/directives/current/part2/index.xhtml>

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

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

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

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

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

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

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

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

## 162 3 Terms and definitions

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

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

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

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

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

### 178 3.1 179 Destination Physical Address

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

### 182 3.2 Dword

183 a 32-bit field

### 184 3.3 185 Flit Mode

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

### 189 3.4 190 MCTP PCIe Endpoint

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

### 192 3.5 193 PCIe Segment

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

### 195 3.6 196 Target Physical Address

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

199



## 200 **4 Symbols and abbreviated terms**

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

### 203 **4.1**

204 **PCIe®**

205 PCI Express™

### 206 **4.2**

207 **VDM**

208 Vendor Defined Message

### 209 **4.3**

210 **CXL™**

211 Compute Express Link™

### 212 **4.4**

213 **FM**

214 Flit Mode

### 215 **4.5**

216 **NFM**

217 Non-Flit Mode

### 218 **4.6**

219 **IDE**

220 Integrity and Data Encryption

### 221 **4.7**

222 **OHC**

223 Orthogonal Header Content

### 224 **4.8**

225 **TLP**

226 Transaction Layer Packet

## 227 **5 Conventions**

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

### 229 **5.1 Reserved and unassigned values**

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

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

### 234 **5.2 Byte ordering**

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

## 237 **6 MCTP over PCI Express VDM transport**

### 238 **6.1 Overview**

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

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

246 The MCTP over PCI Express (PCIe) VDM transport binding transfers MCTP messages using PCIe Type  
247 1 VDMs with data. MCTP messages use the MCTP VDM code value (0000b) stored in the PCIe TAG  
248 field that uniquely differentiates MCTP messages from other DMTF VDMs.

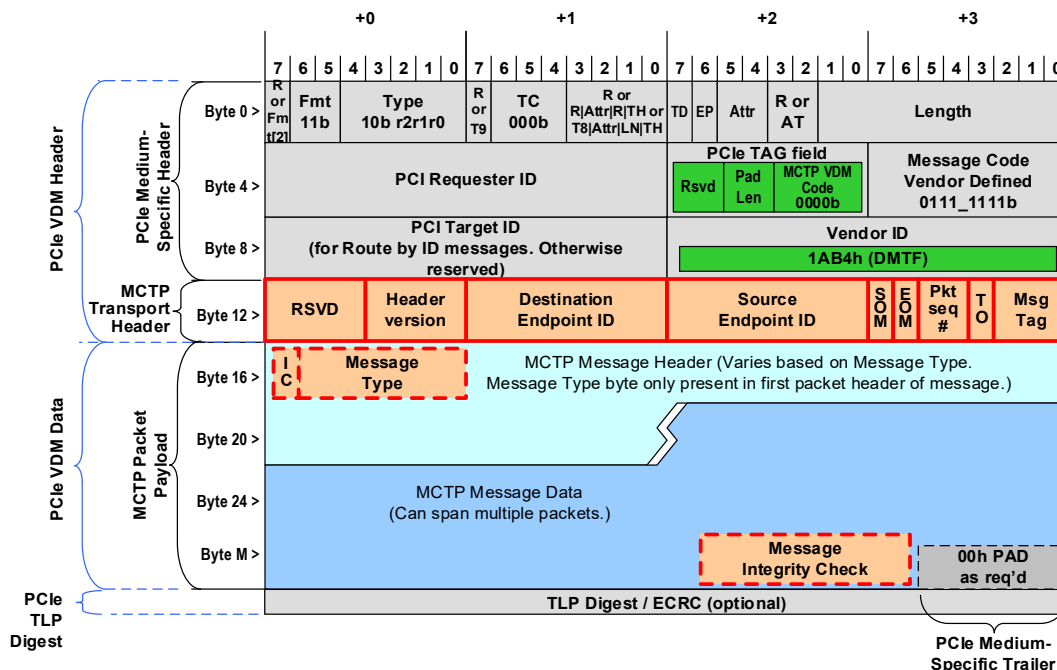
249 Any MCTP baseline transmission unit for MCTP PCIe VDM communication shall be dword aligned.

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

### 252 **6.2 Packet formats**

#### 253 **6.2.1 Non-Flit Mode**

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



255

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

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

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

268 **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.

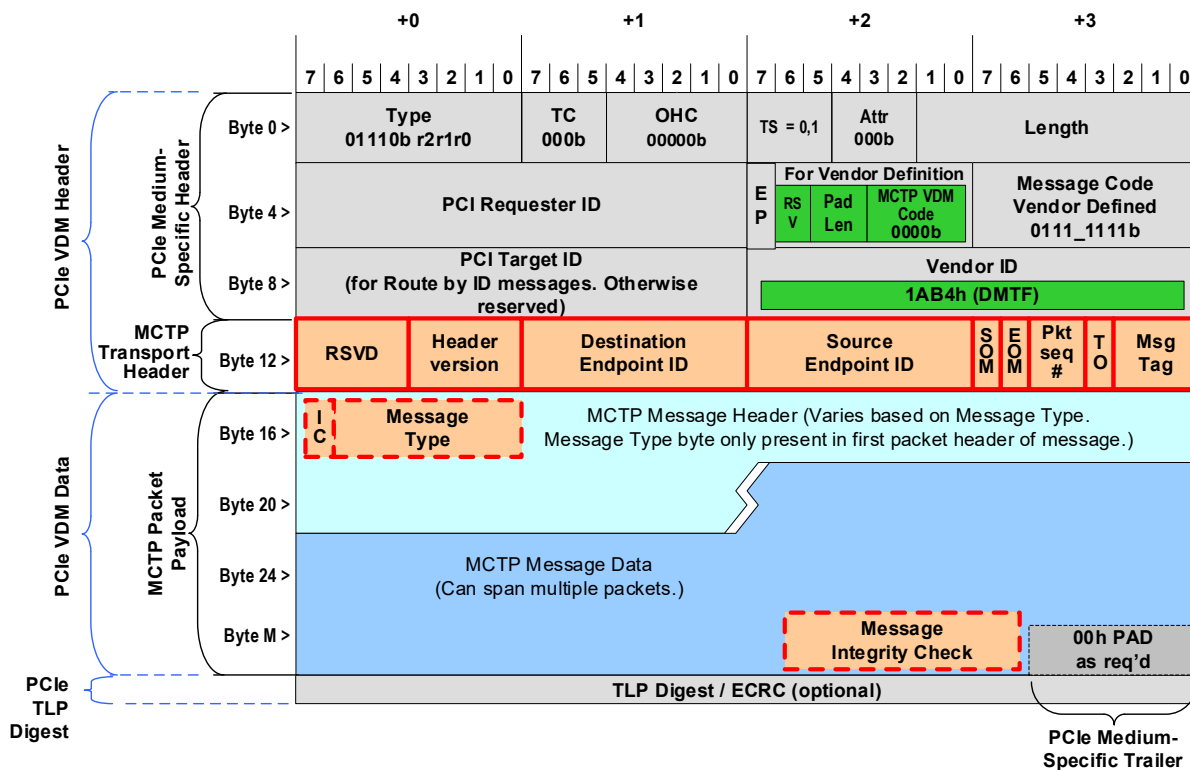
Field	Description
00h PAD	Pad bytes. 0 to 3 bytes of 00h as required to fill out the overall PCIe VDM data to be an integral number of dwords. Because only packets with the EOM bit set to 1b are allowed to be less than the transfer unit size, packets that have the EOM bit set to 0b will already be dword 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.

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

271 **6.2.2 Flit Mode**

272 **Figure 2 and**

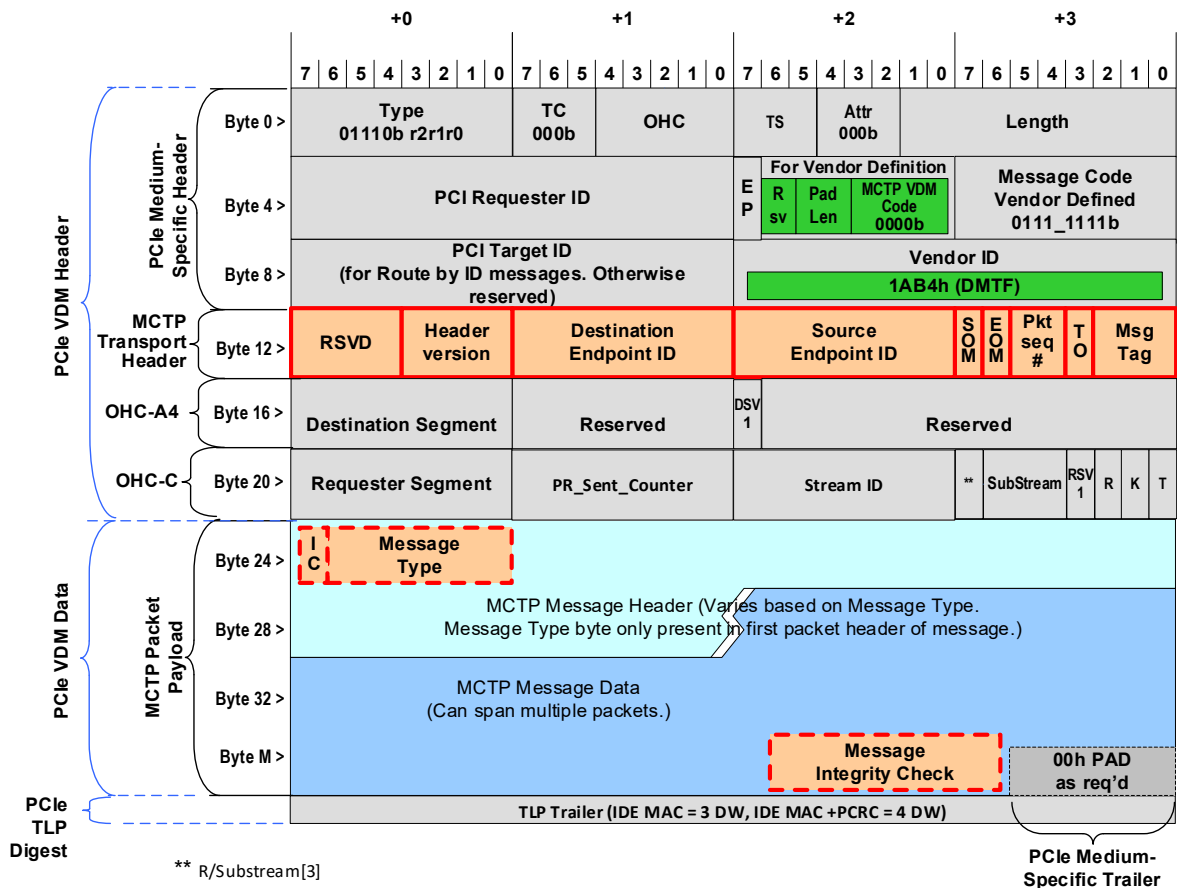
273 Figure 3 show the encapsulation of MCTP packet fields within a PCIe VDM in Flit Mode with or without  
 274 OHCs.



275

276 **Figure 2 – MCTP over PCI Express Vendor Defined Message (VDM) Packet Format**  
 277 **(Flit Mode within a Segment without IDE)**

278



279  
280

281  
282

**Figure 3 – MCTP over PCI Express Vendor Defined Message (VDM) Packet Format (Flit Mode across segments and/or with IDE)**

283 The fields labeled “PCIe Medium-Specific Header” and “PCIe Medium-Specific Trailer” are specific to  
 284 carrying MCTP packets using PCIe VDMs. The fields labeled “MCTP Transport Header” and “MCTP  
 285 Packet Payload” are common fields for all MCTP packets and messages and are specified in [MCTP](#). This  
 286 section defines the location of those fields when they are carried in a Flit Mode PCIe VDM. The PCIe  
 287 specification allows the last four bytes of the PCIe VDM header base to be vendor defined. The MCTP  
 288 over PCIe VDM transport binding specification uses these bytes for MCTP Transport header fields under  
 289 the DMTF Vendor ID. This document also specifies the *medium-specific* use of the MCTP “Hdr Version”  
 290 field.

291 Table 2 lists the PCIe medium-specific fields and field values that shall be used in MCTP over PCIe VDM  
 292 communications. When not specified, field values shall be set according to PCIe specifications. Note that  
 293 the presence of additional OHCs in MCTP over PCIe VDM packets is implementation dependent and  
 294 outside the scope of this specification.

**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.

Field	Description
<b>MCTP Header</b>	
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

### 296 6.3 Supported media

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

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

302

**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

303 Note: The compatibility mentioned above is to a specific PCIe specification version and not to a specific  
304 speed. For example, a device reporting compatibility to PCIe 4.X may still operate in Gen3 or lower  
305 speeds.



306 **6.4 Physical address format for MCTP control messages**

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

311 **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

312 **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

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

317 **6.5 Message routing**

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

322 • **Route by ID**

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

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

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

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

- 331       •   **Route to Root Complex (RC)**
- 332           MCTP endpoints shall use this routing for the Discovery Notify request message to the MCTP  
333           bus owner as part of the MCTP over PCIe VDM discovery process.
- 334           The MCTP endpoints shall use this routing for responding to the MCTP control request  
335           messages that were sent using Broadcast from Root Complex.
- 336           Communication of MCTP PCIe VDM packets that are destined to MCTP bus owner using Route  
337           to Root Complex is implementation specific and is outside the scope of this specification. A  
338           receiver of an MCTP PCIe VDM message with this routing may ignore the Destination EID.
- 339       •   **Broadcast from Root Complex (RC)**
- 340           The MCTP bus owner should use this routing for the Prepare for Endpoint Discovery and  
341           Endpoint Discovery messages as part of the MCTP over PCIe VDM discovery process. If any  
342           other MCTP PCIe VDM message is received with this routing, then the message should be  
343           dropped. Any MCTP PCIe VDM message using this routing should use a destination EID of  
344           0xFF. A receiver of an MCTP PCIe VDM message with this routing may ignore the Destination  
345           EID.

346

### 347   **6.5.1 Routing peer transactions on a PCIe bus**

348   Because the PCIe specification does not require peer-to-peer routing support in PCIe root complexes,  
349   MCTP over PCIe VDM messages are not required to be routed to peer devices directly. When peer-to-  
350   peer routing is not supported by a PCIe root complex, all MCTP over PCIe VDM messages between two  
351   MCTP endpoints shall be routed to or through the MCTP bus owner as an MCTP bridge. If the PCIe root  
352   complex, as the MCTP bus owner, supports peer-to-peer routing, it shall use direct physical addressing to  
353   support routing between two MCTP endpoints on the PCIe bus.

### 354   **6.5.2 Routing messages between PCIe and other buses**

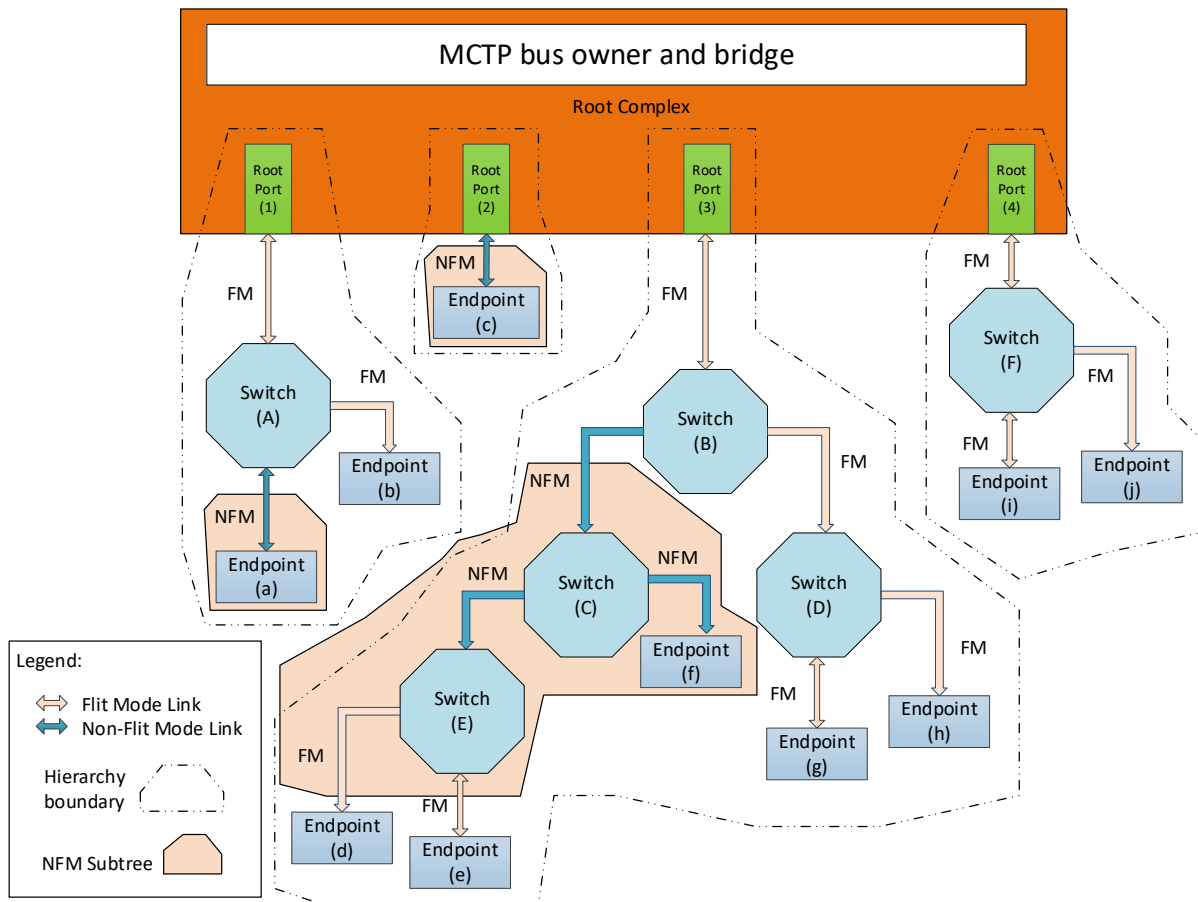
355   All MCTP messages that span between PCIe and other buses shall be sent through the MCTP bus  
356   owner. The MCTP bus owner has the destination EID routing tables necessary to route messages  
357   between the two bus segments.

358   If an endpoint is aware of multiple routes to a destination over multiple bus types, a higher level  
359   algorithm/protocol above MCTP shall be used to determine which bus/route to use. Typically this decision  
360   can be based on things like power state and MCTP discovery state.

### 361   **6.5.3 Routing Messages in Between Flit Mode and Non-Flit Mode domains**

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

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



366

367

Figure 4 – Flit Mode to Non-Flit Mode routing options

368  
369**Table 6 – Address Translations when transitioning between Flit Mode and Non-Flit Mode hierarchies**

Scenario	Fields translation	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)
Flit Mode to Non-Flit Mode across segments	Route to segment indicated by Destination Segment. Target ID kept as is. Requester ID in destination segment = Bridge ID (modified by MCTP bridge)	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. Target ID based on 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. Target ID based on 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 <a href="#">PCIe base spec</a>	No	(e) to (d)

370

371 When a message crosses the boundary between Flit Mode and Non-Flit Mode hierarchies, the Segment  
372 part of the address may be removed (Flit Mode to Non-Flit Mode) or added (Non-Flit Mode to Flit Mode).  
373 Table 6 describes the translations done when a message crosses between Flit Mode and Non-Flit Mode  
374 hierarchies in various scenarios. The “Requires MCTP Bridge” column in Table 6 indicates whether the  
375 described scenario requires one or more PCIe Root Ports to function as an MCTP bridge, as [DSP0236](#)

376 describes. The “Example” column in Table 6 provides an example of the described scenario using the  
 377 topology shown in Figure 4.

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

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

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

389 **6.5.4 Example of Resolve Endpoint ID response**

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

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

Requester	Requester target	Responding MCTP bridge	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 (f)
j	g	Root port 4	{S,B,D,F} of (g)
f	g	Root port 3	{B,D,F} of Root Port 3
f	i	Root port 3	{B,D,F} of Root Port 3
e	c	Root port 3	{B,D,F} of Root Port 3
e	i	Root port 3	{B,D,F} of Root Port 3
e	f	Root port 3	{B,D,F} of (f)

393 Notes:

- 394 • The MCTP bridge may choose to return its own address instead of the endpoint address in all cases.  
 395 In this case, all the traffic will be bridged. The chosen routing method is implementation dependent  
 396 and is outside of the scope of this specification.
- 397 • When a {B,D,F} of Root Port is mentioned, it can be any function within the root complex acting as  
 398 the MCTP bridge.

399 **6.6 Bus owner address**

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

## 402 6.7 Bus and Segment address assignment for PCIe

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

## 404 6.8 Host dependencies

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

### 408 • **Reset**

409 Assertion of “Fundamental Reset” on the bus causes both the host functionality as well as the  
410 MCTP PCIe VDM communication on an MCTP PCIe endpoint to be reset. From the assertion  
411 “Fundamental Reset” until a physical address is assigned as part of the PCIe fabric  
412 enumeration, no MCTP over PCIe VDM messages can be sent.

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

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

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

### 422 • **Configuration and enumeration**

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

### 427 • **Power management states**

428 The host (as defined in the context of the [PCI Express™ specification](#)) controls PCIe bus power  
429 management. The host may power down PCIe devices and links, or place them in sleep states,  
430 independent of management controllers, which may cause MCTP PCIe VDM communication to  
431 be unavailable. An MCTP PCIe VDM endpoint on a PCIe device may not be able to respond to  
432 PCIe VDMs in low power states or sleep states. The PCIe device may notify using mechanisms  
433 outside of this specification of the current power state to delay any PCIe VDM communication  
434 until the PCIe device transitions to a state where the PCIe VDM communication is available.  
435 Depending on the device usage in the system, a PCIe device may retain or lose states such as  
436 EID, “discovered” state, and routing information (if the device is an MCTP bridge). A PCIe  
437 device that loses MCTP PCIe VDM communication state needs to be reinitialized and/or  
438 rediscovered after it returns to a power state that supports MCTP over PCIe VDM  
439 communication.

### 440 • **Link states**

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

445       •   **PCIe Root Complex**

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

448   **6.9 Discovery Notify message use for PCIe**

449   An MCTP control Discovery Notify message shall be sent from a PCIe endpoint to the MCTP bus owner  
450   whenever the physical address for the device changes (that is, the endpoint receives a Type 0  
451   configuration write request and the bus number and/or the segment number is different from the currently  
452   stored bus and segment numbers). This occurs on the first Type 0 configuration write following a PCIe  
453   bus reset during initial enumeration, or during re-enumeration where the bus or segment number has  
454   changed (for example, because of a hot plug event, bus reset, and so on).

455   Endpoints use the Discovery Notify command to inform the MCTP bus owner that it needs to update the  
456   endpoint's ID. The Discovery Notify command shall be sent with the PCIe message routing set to 000b  
457   (Route to Root Complex), the Destination Endpoint ID for the Discovery Notify message shall be set to  
458   the Null Destination EID. The Source Endpoint ID field shall be set to the Null Source EID if the device  
459   has not yet been assigned an EID; otherwise, it shall contain the assigned EID value.

460   **6.10 MCTP over PCIe endpoint discovery**

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

462   **6.10.1 Discovered flag**

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

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

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

- 470       •   Whenever the PCIe physical address associated with the endpoint is initially assigned or is  
471       changed to a different value.
- 472       •   Whenever an endpoint first appears on the bus and requires an EID assignment. A device shall  
473       have been enumerated on PCI and have a bus/device/function or bus/function number before it  
474       can do this.
- 475       •   During operation if an endpoint enters a state that causes it to lose its EID assignment.
- 476       •   For endpoints that have already received an EID assignment but are in any temporary state  
477       where the endpoint was unable to respond to MCTP control requests for more than T<sub>RECLAIM</sub>  
478       seconds.

479   Only endpoints that have their Discovered flag set to *undiscovered* shall respond to the Endpoint  
480   Discovery message. Endpoints that have the flag set to *discovered* shall not respond to the Endpoint  
481   Discovery message.

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

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

### 488 6.10.2 PCIe endpoint announcement

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

### 493 6.10.3 Full endpoint Discovery/Enumeration

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

- 496 1) The MCTP bus owner issues a broadcast Prepare for Endpoint Discovery message. This  
497 message causes each discoverable endpoint on the bus to set its PCIe endpoint Discovered  
498 flag to undiscovered. Depending on the number of endpoints and the buffer space available in  
499 the MCTP bus owner, the MCTP bus owner may not receive all of the response messages. The  
500 discovery process does not require the MCTP bus owner to receive all the response messages  
501 to the Prepare for Endpoint Discovery request. Because the MCTP bus owner cannot determine  
502 that all endpoints have received the Prepare for Endpoint Discovery request, it is recommended  
503 that Prepare for Endpoint Discovery request is retried MN1 times to help ensure that all  
504 endpoints have received the request. The MCTP bus owner is not required to wait for MT2 time  
505 interval between the retries.
- 506 2) The MCTP bus owner should wait for MT2 time interval to help ensure that all endpoints that  
507 received the Prepare for Endpoint Discovery request have processed the request.
- 508 3) The MCTP bus owner issues a broadcast Endpoint Discovery request message. All MCTP-  
509 capable devices that have their Discovered flag set to undiscovered will respond with an  
510 Endpoint Discovery response message.
- 511 4) Depending on the number of endpoints and the buffer space available in the MCTP bus owner,  
512 the MCTP bus owner receives some or all of these response messages. For each response  
513 message received from an undiscovered MCTP-capable device PCIe bus/device/function or  
514 bus/function number, the MCTP bus owner issues a Set Endpoint ID command to the physical  
515 address for the endpoint. This causes the endpoint to set its Discovered flag to *discovered*.  
516 From this point, the endpoint shall not respond to the Endpoint Discovery command until  
517 another Prepare for Endpoint Discovery command is received or some other condition causes  
518 the Discovered flag to be set back to *undiscovered*.
- 519 5) If the MCTP bus owner received any responses to the Endpoint Discovery request issued in  
520 Step 3, then it shall repeat steps 3 and 4 until it no longer gets any responses to the Endpoint  
521 Discovery request. In this case, then the MCTP bus owner is allowed to send the next Endpoint  
522 Discovery request without waiting for MT2 time interval. If no responses were received by the  
523 MCTP bus owner to the Endpoint Discovery request within the MT2 time interval, then the  
524 discovery process is completed.

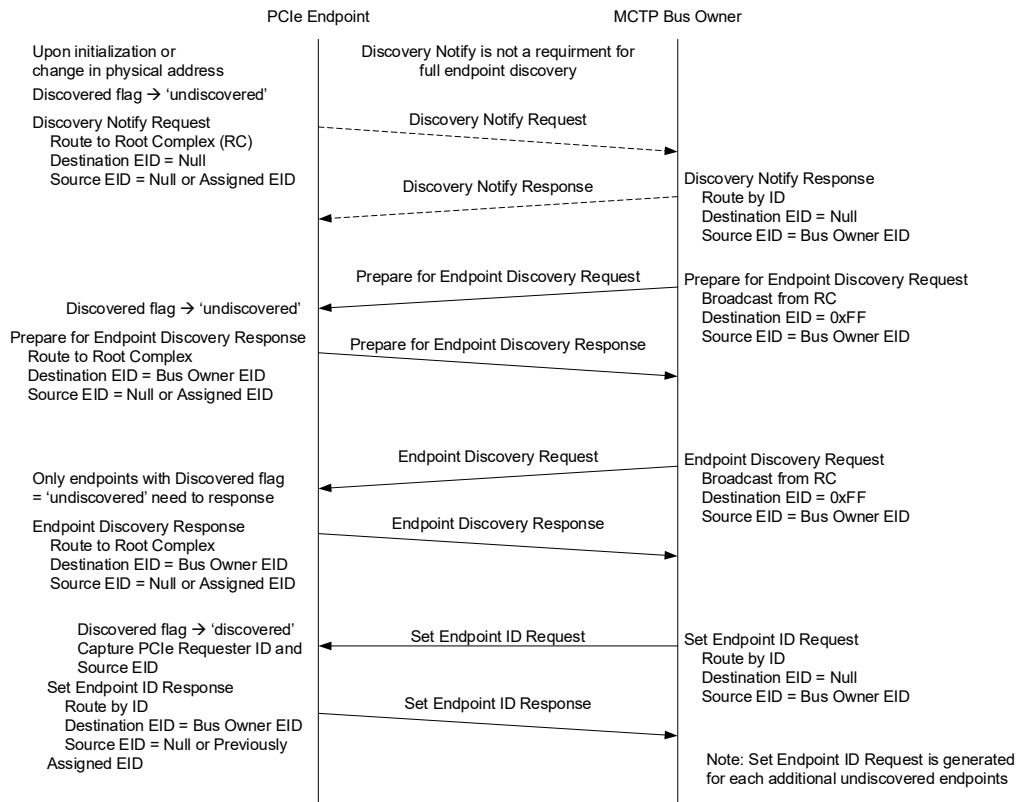
525 After the initial endpoint enumeration, it is recommended that the MCTP bus owner maintains a list of the  
526 unique IDs for the endpoints it has discovered and reassigns the same IDs to those endpoints if a  
527 physical address changes during system operation.

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

530 Figure 5 provides an example flow of operations for full endpoint discovery.



**Full PCIe MCTP Endpoint Discovery Example**



531

532

**Figure 5 – Example Flow of operations for full MCTP Discovery over PCIe**

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

535 **6.10.4 Partial endpoint Discovery/Enumeration**

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

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

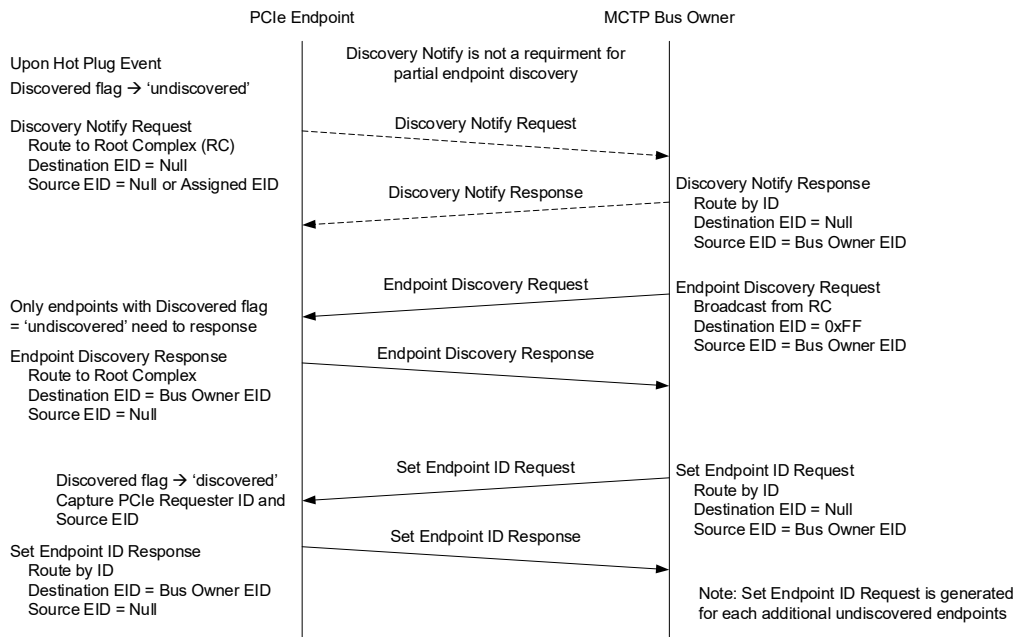
544 The partial discovery process may be initiated when a device that is added or enabled for MCTP sends a  
545 Discovery Notify message to the MCTP bus owner. The MCTP bus owner may also elect to periodically  
546 issue a broadcast Endpoint Discovery message to test for whether any undiscovered endpoints have  
547 been missed. The Discovery Notify message provides the MCTP bus owner with the physical address of  
548 the MCTP PCIe endpoint. The MCTP bus owner can then send a directed Endpoint Discovery message  
549 to the endpoint to confirm that the device has not been discovered. The MCTP bus owner then issues a  
550 Set Endpoint ID command to the physical address for the endpoint which causes the endpoint to set its  
551 Discovered flag to *discovered*.

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

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

556 Figure 6 provides an example flow of operations for partial endpoint discovery.

Partial PCIe MCTP Endpoint Discovery



557

558 **Figure 6 – Flow of operations for Partial Endpoint Discovery**

559 **6.10.5 Endpoint re-enumeration**

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

565 **6.11 MCTP messages timing requirements**

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

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 <sup>[1]</sup> + 6 ms	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.
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.
NOTE 1: Unless otherwise specified, this timing applies to the mandatory and optional MCTP commands.				
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.				

## ANNEX A (informative)

570  
571  
572  
573

### Notations and conventions

#### 574 Notations

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

- 576 • 2:N In field descriptions, this will typically be used to represent a range of byte offsets  
577 starting from byte two and continuing to and including byte N. The lowest offset is on  
578 the left, the highest is on the right.
- 579 • (6) Parentheses around a single number can be used in message field descriptions to  
580 indicate a byte field that may be present or absent.
- 581 • (3:6) Parentheses around a field consisting of a range of bytes indicates the entire range  
582 may be present or absent. The lowest offset is on the left, the highest is on the right.
- 583 • [PCIe](#) Underlined, blue text is typically used to indicate a reference to a document or  
584 specification called out in Clause 2, "Normative References" or to items hyperlinked  
585 within the document.
- 586 • rsvd Abbreviation for Reserved. Case insensitive.
- 587 • [4] Square brackets around a number are typically used to indicate a bit offset. Bit offsets  
588 are given as 0-based values (that is, the least significant bit [LSb] offset = 0).
- 589 • [7:5] A range of bit offsets. The most significant bit is on the left, the least significant bit is  
590 on the right.
- 591 • 1b The lower case "b" following a number consisting of 0s and 1s is used to indicate the  
592 number is being given in binary format.
- 593 • 0x12A A leading "0x" is used to indicate a number given in hexadecimal format.

594  
595  
596  
597  
598

## 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>

599