



1
2
3
4

Document Identifier: DSP2061

Date: 2024-09-19

Version: 1.0.1

5 **PLDM Accelerator Modeling**

6 **Supersedes: 1.0.0**
7 **Document Class: Informative**
8 **Document Status: Published**
9 **Document Language: en-US**

10 Copyright Notice

11 Copyright © 2022, 2024 DMTF. All rights reserved.

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

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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

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

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

CONTENTS

| | | |
|----|--|----|
| 37 | Foreword | 7 |
| 38 | Introduction..... | 8 |
| 39 | Document conventions..... | 8 |
| 40 | Typographical conventions | 8 |
| 41 | ABNF usage conventions | 8 |
| 42 | 1 Scope | 9 |
| 43 | 2 Normative references | 9 |
| 44 | 3 Terms and definitions | 10 |
| 45 | 4 Symbols and abbreviated terms..... | 11 |
| 46 | 5 PLDM Accelerator Modeling overview | 12 |
| 47 | 5.1 General | 12 |
| 48 | 5.2 Model elements..... | 12 |
| 49 | 5.2.1 PLDM terminus | 12 |
| 50 | 5.2.2 Accelerator card..... | 12 |
| 51 | 5.2.3 Accelerator | 13 |
| 52 | 5.2.4 Memory | 13 |
| 53 | 5.2.5 Inter-Accelerator card connection..... | 13 |
| 54 | 5.3 Model sensors..... | 13 |
| 55 | 5.3.1 General | 13 |
| 56 | 5.3.2 Accelerator card temperature sensor | 14 |
| 57 | 5.3.3 Accelerator card power sensor | 14 |
| 58 | 5.3.4 Accelerator card fan speed sensor | 14 |
| 59 | 5.3.5 Accelerator card voltage sensor | 14 |
| 60 | 5.3.6 Accelerator card auxiliary device temperature sensor..... | 15 |
| 61 | 5.3.7 Accelerator card auxiliary device health sensor | 15 |
| 62 | 5.3.8 Accelerator card composite state sensor | 15 |
| 63 | 5.3.9 Accelerator temperature sensor | 15 |
| 64 | 5.3.10 Accelerator power sensor | 15 |
| 65 | 5.3.11 Accelerator composite state sensor..... | 15 |
| 66 | 5.3.12 Accelerator clock speed sensor..... | 16 |
| 67 | 5.3.13 Memory temperature sensor..... | 16 |
| 68 | 5.3.14 Memory error statistics | 16 |
| 69 | 5.3.15 Memory composite state sensor..... | 16 |
| 70 | 5.4 Hierarchy description of the Accelerator card model elements | 16 |
| 71 | 5.4.1 General | 16 |
| 72 | 5.4.2 Physical entities association | 16 |
| 73 | 5.4.3 Logical entity association..... | 17 |
| 74 | 5.4.4 Sensor association..... | 18 |
| 75 | 5.5 Element PLDM Type IDs | 19 |
| 76 | 5.6 Enumeration..... | 19 |
| 77 | 5.6.1 General | 19 |
| 78 | 5.6.2 Enumeration scheme | 20 |
| 79 | 5.7 Model illustration | 22 |
| 80 | 5.7.1 General | 22 |
| 81 | 5.7.2 Accelerator Card | 22 |
| 82 | 5.7.3 Accelerator | 22 |
| 83 | 5.7.4 Memory | 22 |
| 84 | 5.8 Events..... | 22 |
| 85 | 5.8.1 General | 22 |
| 86 | 5.8.2 Accelerator firmware version change | 22 |
| 87 | 5.8.3 Health and state sensors events notifications | 22 |
| 88 | 6 Model use example | 23 |

| | | | |
|-----|-----------------------|---|----|
| 89 | 6.1 | General | 23 |
| 90 | 6.2 | Model hierarchy | 25 |
| 91 | 6.3 | Top-level TID | 25 |
| 92 | 6.4 | Accelerator card..... | 26 |
| 93 | 6.4.1 | General | 26 |
| 94 | 6.4.2 | Accelerator card power sensor | 28 |
| 95 | 6.4.3 | Accelerator card temperature sensor | 28 |
| 96 | 6.4.4 | Accelerator card fan speed sensor | 29 |
| 97 | 6.4.5 | Accelerator card voltage sensor | 29 |
| 98 | 6.4.6 | Accelerator card auxiliary device temperature sensor..... | 30 |
| 99 | 6.4.7 | Accelerator card auxiliary device health sensor | 30 |
| 100 | 6.4.8 | Accelerator card composite state sensor | 31 |
| 101 | 6.5 | Accelerator | 32 |
| 102 | 6.5.1 | General | 32 |
| 103 | 6.5.2 | Accelerator temperature sensor | 33 |
| 104 | 6.5.3 | Accelerator power sensor | 33 |
| 105 | 6.5.4 | Accelerator composite state sensor..... | 34 |
| 106 | 6.5.5 | Accelerator clock speed sensor..... | 35 |
| 107 | 6.6 | Memory | 35 |
| 108 | 6.6.1 | General | 35 |
| 109 | 6.6.2 | Memory temperature sensor..... | 36 |
| 110 | 6.6.3 | Memory error statistics sensors..... | 36 |
| 111 | 6.6.4 | Memory composite state sensor | 37 |
| 112 | ANNEX A (informative) | Notation and conventions | 38 |
| 113 | ANNEX B (informative) | Change log..... | 39 |
| 114 | | | |

115 **Figures**

116 Figure 1 – Inter-Accelerator card connection 13

117 Figure 2 – Accelerator card PLDM model diagram 14

118 Figure 3 – Hierarchy description using ContainerEntityContainerID referencing the Container Entity

119 ContainerID 17

120 Figure 4 – Defining a logical association 18

121 Figure 5 – Top-level sensor association 19

122 Figure 6 – Example model diagram 24

123 Figure 7 – Accelerator card model hierarchy 25

124 Figure 8 – Accelerator card level elements 26

125 Figure 9 – Accelerator card container PDR 27

126 Figure 10 – Accelerator card power sensor PDR 28

127 Figure 11 – Ambient Temperature sensor PDR 28

128 Figure 12 – Accelerator card fan speed sensor PDR 29

129 Figure 13 – Accelerator card voltage sensor PDR 29

130 Figure 14 – Auxiliary device temperature sensor PDR 30

131 Figure 15 – Auxiliary device health sensor PDR 30

132 Figure 16 – Accelerator card composite state sensor PDR 31

133 Figure 17 – Example model Accelerator 32

134 Figure 18 – Accelerator entity association PDR 32

135 Figure 19 – Accelerator temperature sensor PDR 33

136 Figure 20 – Accelerator power sensor PDR 33

137 Figure 21 – Accelerator composite state sensor PDR 34

138 Figure 22 – Accelerator card clock speed sensor PDR 35

139 Figure 23 – Example Memory model 35

140 Figure 24 – Memory association PDR 36

141 Figure 25 – Memory temperature sensor PDR 36

142 Figure 26 – Memory correctable errors PDR 36

143 Figure 27 – Memory uncorrectable errors PDR 37

144 Figure 28 – Memory composite state sensor PDR 37

145

146 **Tables**

147 Table 1 – Type IDs used in the Accelerator card model..... 19
148 Table 2 – Chosen enumeration limits in the model..... 20
149 Table 3 – Example Enumeration Scheme with Type IDs 21
150 Table 4 – TID PDR..... 25
151

152

Foreword

153 The PLDM Accelerator Modeling (DSP2061) document was prepared by the Platform Management
154 Communications Infrastructure (PMCI) Working Group of DMTF.

155 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems
156 management and interoperability. For information about DMTF, see <https://www.dmtf.org>.

157 **Acknowledgments**

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

159 **Editors:**

- 160 • Rama Rao Bisa – Dell Technologies
- 161 • Pavan Kumar Gavvala – Dell Technologies
- 162 • Eliel Louzoun – Intel Corporation

163 **Contributors:**

- 164 • Patrick Caporale – Lenovo
- 165 • Michael Garner – Meta
- 166 • Yuval Itkin – NVIDIA Corporation
- 167 • Deepak Kodihalli – NVIDIA Corporation
- 168 • Hemal Shah – Broadcom Inc.
- 169 • Bob Stevens – Dell Technologies
- 170 • Pierre-Philippe Stevens – Advanced Micro Devices
- 171 • Ryan Weldon – Groq

172

Introduction

173 This document describes a modeling scheme for an Accelerator card using PLDM for Platform Monitoring
174 and Control [DSP0248](#) semantics.

175 Document conventions

176 Typographical conventions

177 The following typographical conventions are used in this document:

- 178 • Document titles are marked in *italics*.
- 179 • Important terms that are used for the first time are marked in *italics*.
- 180 • Terms include a link to the term definition in the "Terms and definitions" clause, enabling easy
181 navigation to the term definition.
- 182 • ABNF rules are in `monospaced font`.

183 ABNF usage conventions

184 Format definitions in this document are specified using ABNF (see [RFC 5234](#)), with the following
185 deviations:

- 186 • Literal strings are to be interpreted as case-sensitive Unicode characters, as opposed to the
187 definition in [RFC 5234](#) that interprets literal strings as case-insensitive US-ASCII characters.

188 Reserved and unassigned values

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

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

193 Byte ordering

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

196 Other Conventions

197 See ANNEX A for other conventions.

198

PLDM Accelerator Modeling

199 1 Scope

200 This document defines an example data model for implementing the systems management of
201 accelerators using PLDM for Platform Monitoring and Control [DSP0248](#) semantics. This document
202 establishes a common framework that can provide implementation consistency between a system's
203 Management Controller and accelerators and accelerator cards the system contains, focusing on FPGAs
204 and GPUs and similar devices that offload processing from the host CPU. This data model is assumed to
205 be extensible to a variety of physical implementations and should not be construed to be limited to the
206 examples herein.

207 Accelerators and Accelerator card implementations may include ancillary features such as networking
208 and storage that have management schemas defined in other data models and Specifications. The
209 management of those features is outside the scope of this data model. The data model provided here
210 focuses on the management of the accelerator features of the card, but composite sensors that return
211 overall card status for example, may include metadata from those other functional areas. For instance, it
212 may be appropriate to use either [DSP2054](#) or [DSP0222](#) for the management of networking features that
213 may be included on the accelerator or card.

214 2 Normative references

215 The following referenced documents are indispensable for the application of this document. For dated or
216 versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies.
217 For references without a date or version, the latest published edition of the referenced document
218 (including any corrigenda or DMTF update versions) applies.

219 Unless otherwise specified, for DMTF documents this means any document version that has minor or
220 update version numbers that are later than those for the referenced document. The major version
221 numbers must match the major version number given for the referenced document.

222 DMTF DSP0222, *Network Controller Sideband Interface (NC-SI) Specification 1.1*,
223 https://www.dmtf.org/sites/default/files/standards/documents/DSP0222_1.1.pdf

224 DMTF DSP0236, *Management Component Transport Protocol (MCTP) Base Specification 1.3*,
225 https://www.dmtf.org/sites/default/files/standards/documents/DSP0236_1.3.pdf

226 DMTF DSP0240, *Platform Level Data Model (PLDM) Base Specification 1.1*,
227 https://www.dmtf.org/sites/default/files/standards/documents/DSP0240_1.1.pdf

228 DMTF DSP0241, *Platform Level Data Model (PLDM) Over MCTP Binding Specification 1.0*,
229 https://www.dmtf.org/sites/default/files/standards/documents/DSP0241_1.0.pdf

230 DMTF DSP0245, *Platform Level Data Model (PLDM) IDs and Codes Specification 1.3*,
231 https://www.dmtf.org/sites/default/files/standards/documents/DSP0245_1.3.pdf

232 DMTF DSP0248, *Platform Level Data Model (PLDM) for Platform Monitoring and Control Specification*
233 *1.2*, https://www.dmtf.org/sites/default/files/standards/documents/DSP0248_1.2.pdf

234 DMTF DSP0249, *Platform Level Data Model (PLDM) State Set Specification 1.1*,
235 https://www.dmtf.org/sites/default/files/standards/documents/DSP0249_1.1.pdf

236 DMTF DSP0257, *Platform Level Data Model (PLDM) FRU Data Specification 1.0*,
237 https://www.dmtf.org/sites/default/files/standards/documents/DSP0257_1.0.pdf

- 238 DMTF DSP0267, *Platform Level Data Model (PLDM) for Firmware Update Specification 1.1*,
239 https://www.dmtf.org/sites/default/files/standards/documents/DSP0267_1.1.pdf
- 240 DMTF DSP2054, *PLDM NIC Modeling 1.0*,
241 https://dmft.org/sites/default/files/standards/documents/DSP2054_1.0.pdf
- 242 IETF RFC 2781, *UTF-16, an encoding of ISO 10646*, February 2000,
243 <https://www.ietf.org/rfc/rfc2781.txt>
- 244 IETF STD 63, *UTF-8, a transformation format of ISO 10646*, <https://www.ietf.org/rfc/std/std63.txt>
- 245 IETF RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace*, July 2005,
246 <https://www.ietf.org/rfc/rfc4122.txt>
- 247 IETF RFC 4646, *Tags for Identifying Languages*, September 2006,
248 <https://www.ietf.org/rfc/rfc4646.txt>
- 249 IETF RFC 5234, *Augmented BNF for Syntax Specifications: ABNF*, January 2008,
250 <https://datatracker.ietf.org/doc/html/rfc5234>
- 251 ISO 8859-1, *Final Text of DIS 8859-1, 8-bit single-byte coded graphic character sets — Part 1: Latin
252 alphabet No. 1*, February 1998
- 253 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of ISO and IEC documents*,
254 <https://www.iso.org/sites/directives/current/part2/index.xhtml>

255 **3 Terms and definitions**

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

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

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

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

269 Refer to [DSP0240](#) for terms and definitions that are used across the PLDM specifications.

270 **4 Symbols and abbreviated terms**

271 Refer to [DSP0240](#) and [DSP0248](#) for symbols and abbreviated terms that are used across the PLDM
272 specifications. For the purposes of this document, the following additional symbols and abbreviated terms
273 apply.

274 **4.1**

275 **PCB**

276 Printed Circuit Board

277 **4.2**

278 **FPGA**

279 Field Programmable Gate Array

280 **4.3**

281 **GPU**

282 Graphics Processing Unit

283 5 PLDM Accelerator Modeling overview

284 5.1 General

285 This document describes a hierarchical modeling scheme for an Accelerator card using PLDM for
286 Platform Monitoring and Control [DSP0248](#) semantics. The model is scalable, allowing consistent
287 modeling of Accelerator cards with different configuration options such as the number of Accelerators.

288

289 While PLDM for Platform Monitoring and Control [DSP0248](#) is a published standard, using the model
290 defined in this document simplifies interoperability by establishing a consistent schema.

291

292 The basic format that is used for sending PLDM messages is defined in [DSP0240](#). The format that is
293 used for carrying PLDM messages over a transport-layer protocol and medium is given in companion
294 documents to the base specification. For example, [DSP0241](#) defines how PLDM messages are formatted
295 and sent using MCTP as the transport.

296 The model supports the following:

- 297 • Consistent modeling of an Accelerator card regardless of the specific configuration and resource
298 count
- 299 • Accelerator card hardware structure description
- 300 • Reporting of configuration changes such as firmware update

301 5.2 Model elements

302 5.2.1 PLDM terminus

303 PLDM for Platform Monitoring and Control [DSP0248](#) defines a single root for every device instance,
304 referred to as PLDM Terminus and identified with a TID. The term “MC” is used to identify a PLDM
305 terminus which communicates with an Accelerator card throughout this document.

306 When there are multiple Accelerators assembled on the same card, there may be a single Accelerator
307 which reports all the sensors of all the elements on the Accelerator card to the MC. Alternatively, each
308 Accelerator in the Accelerator card may present a separate PLDM terminus.

309 PLDM for Platform Monitoring and Control [DSP0248](#) does not allow associating components reported via
310 different PLDM termini since every database is relative to a given PLDM terminus. To overcome this
311 constraint, the implementers can retrieve a globally unique ID (Board part number and serial number)
312 from each TID and recognize these TIDs belonging to the same Accelerator card. The process to retrieve
313 the globally unique ID (Board part number and serial number) from each TID is outside of this document.

314 All PLDM IDs specified by the model in this document shall be consistent across all TIDs on a given card.
315 This avoids conflict from duplication of IDs in the combined model, generated by merging the TID-specific
316 model elements reported as part of the overall model.

317 5.2.2 Accelerator card

318 In this model, the Accelerator card is the top-level element of the hierarchy containing one or more
319 Accelerators on a PCB. An Accelerator card is a hardware and software solution that offloads certain
320 processing from the host processor. The Accelerator card in this document refers to various form factors
321 and is represented with PLDM Entity ID code 68 for Add-in card. The Accelerator card may contain
322 sensors.

323 **5.2.3 Accelerator**

324 In this model, an Accelerator is the second level element of the hierarchy containing one or more sensors.
 325 An Accelerator is a hardware device with a main function of offloading certain processing from the host
 326 processor. An Accelerator may contain sensors such as health state, power-consumption, and
 327 temperature.

328 **5.2.4 Memory**

329 The term memory in this document covers the internal memory of the Accelerator, memory chips installed
 330 on the PCB and the DIMMs. In this model, the memory is at the second level of the hierarchy. A Memory
 331 may contain sensors such as temperature, health state, and error statistics.

332 **5.2.5 Inter-Accelerator card connection**

333 The Accelerator cards may support communication with each other. Figure 1 depicts an Inter-Accelerator
 334 card connection, and it may not be the only communication interface between Accelerator cards.

335

336

337

338

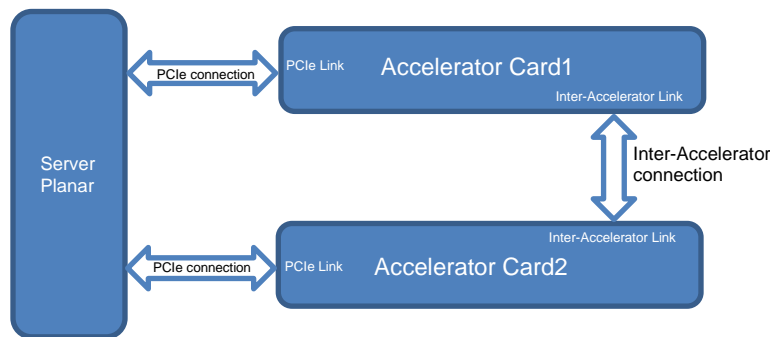
339

340

341

342

343



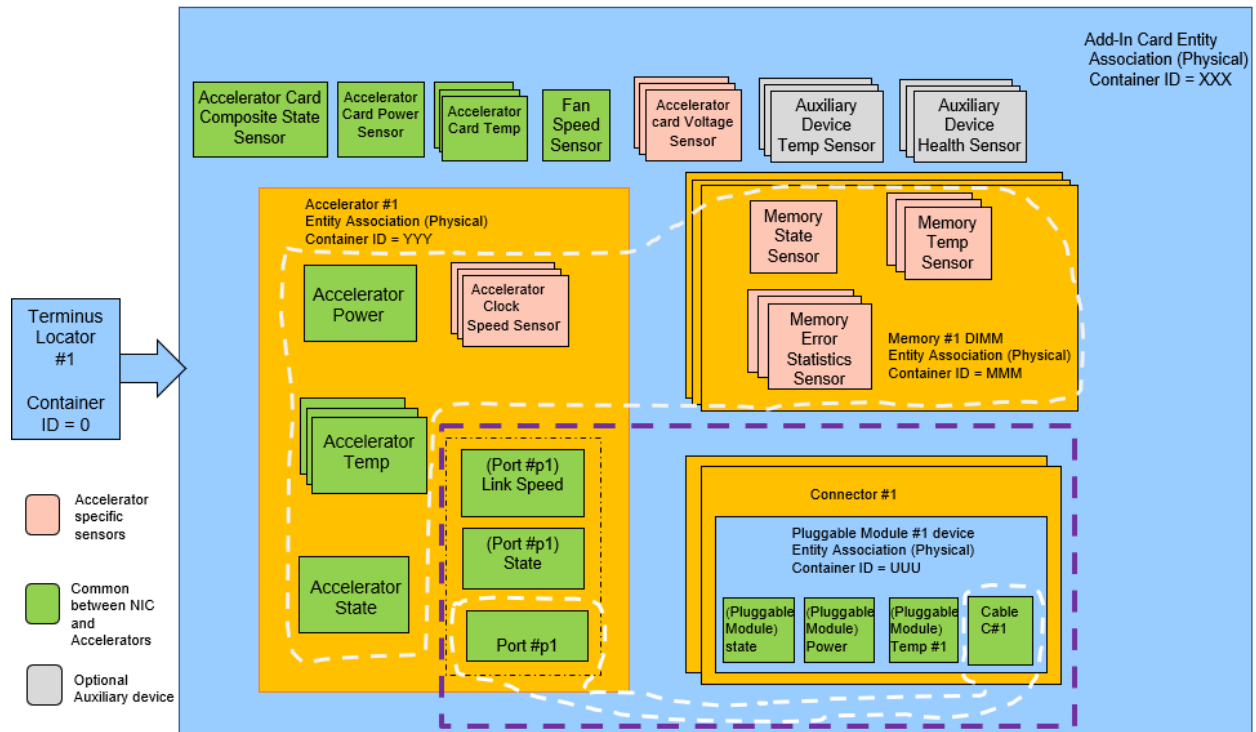
344 **Figure 1 – Inter-Accelerator card connection**

345 **5.3 Model sensors**

346 **5.3.1 General**

347 Attributes are reported by means of sensors. Numeric sensors are used to report specific measured
 348 attributes. State sensors report operational and/or health state. The default thresholds for all numeric
 349 sensors shall be set by the hardware vendor. The sensors can be associated with any entity such as the
 350 Accelerator card, Accelerator or Memory. The description of each sensor is applicable only for the
 351 implemented sensors and it is not mandatory to implement all the sensors described in this document.
 352 There may be auxiliary devices present on the accelerator card and each auxiliary device may present its
 353 own set of sensors.

354 The Sensor Auxiliary Names PDR is recommended to provide the proper name of each sensor.



355

356

Figure 2 – Accelerator card PLDM model diagram

357 **5.3.2 Accelerator card temperature sensor**

358 The temperature sensor on the Accelerator card reports the card’s ambient temperature and is
 359 represented using a numeric sensor. There may be multiple temperature sensors installed on the
 360 Accelerator card.

361 **5.3.3 Accelerator card power sensor**

362 The power sensor on the Accelerator card reports the estimated or measured aggregate power
 363 consumption of the Accelerator card and is represented using a numeric sensor. An Accelerator card
 364 which cannot accurately report its real-time power consumption may report its estimated maximal power.
 365 When there are multiple Accelerators on the same Accelerator card, there may be no visibility by any
 366 Accelerator to the real-time information of the other Accelerators. For this reason, this sensor is only
 367 implemented when there is only one Accelerator on the Accelerator card, or when there is a hardware
 368 sensor which does allow measuring and reporting the total card power consumption or when the maximal
 369 estimated power is reported without being measured or when the accelerators can communicate with
 370 each other.

371 **5.3.4 Accelerator card fan speed sensor**

372 The fan speed sensor on the Accelerator card reports the speed of an active cooling fan and is
 373 represented using a numeric sensor. An Accelerator card may have multiple fans installed, each
 374 potentially with its own speed sensor.

375 **5.3.5 Accelerator card voltage sensor**

376 The voltage sensors on the Accelerator card report various voltages on the card and are represented
 377 using numeric sensors. There may be multiple voltage sensors installed on the card.

378 **5.3.6 Accelerator card auxiliary device temperature sensor**

379 The temperature sensor on the auxiliary device reports the ambient temperature of the auxiliary device
380 and is represented using a numeric sensor. This document does not mandate having an auxiliary device
381 temperature sensor.

382 **5.3.7 Accelerator card auxiliary device health sensor**

383 The health sensor on the auxiliary device reports the health state of the auxiliary device and is
384 represented using a state sensor. This document does not mandate having an auxiliary device health
385 sensor.

386 **5.3.8 Accelerator card composite state sensor**

387 The Accelerator card composite state sensor combines the Accelerator card thermal state sensor, the
388 Memory operational fault state sensor, and the Accelerator card health state sensor. The Accelerator card
389 health state is the aggregated health state of all the components on the card. The reported aggregated
390 health state of the Accelerator card reflects the worst case of the reported health states for each of the
391 elements monitored in the model. For example, if an Accelerator health state is non-critical and a memory
392 health state is critical, then the Accelerator card health state may be set to critical in the Accelerator card
393 composite state sensor.

394 When there are multiple Accelerators, there may be no visibility by any Accelerator to the real-time
395 information of other Accelerators. For this reason, this composite state sensor is only implemented when
396 there is only a single Accelerator on the Accelerator card or when the Accelerator card has the needed
397 visibility of all the components such as Accelerators and memory.

398 To determine the respective sensor states, the following steps shall be used: the accelerator card thermal
399 state sensor shall also reflect the auxiliary device temperature and the accelerator card health state
400 sensor shall also reflect the auxiliary device health state.

401 **5.3.9 Accelerator temperature sensor**

402 The temperature sensor of the Accelerator reflects the device temperature and is represented using a
403 numeric sensor. This sensor is typically located in the thermally sensitive areas on the Accelerator.

404 **5.3.10 Accelerator power sensor**

405 The power sensor on the Accelerator reports the estimated or measured power consumption of the
406 Accelerator and represented using a numeric sensor. An Accelerator which cannot accurately report its
407 real-time power consumption may report its estimated maximal power.

408 **5.3.11 Accelerator composite state sensor**

409 The Accelerator composite state sensor combines the Accelerator Thermal trip state, Accelerator health
410 state, Configuration valid state, Configuration change state, and Accelerator firmware version change
411 state. The MC can use this sensor to identify issues with the Accelerator and to identify the specific
412 maintenance operations that it needs to perform. These operations may include Accelerator reset,
413 system-level shutdown for thermal protection, and other system-level maintenance.

414 Using the configuration change indication, the Accelerator notifies the MC to retrieve PDRs updated by
415 the configuration change.

416 When a firmware update is detected, the composite state sensor can reflect this event to the MC, allowing
417 the MC to take any action needed to respond to the update. Note that reading the new firmware version
418 may be performed by the MC using protocols other than PLDM for Platform Monitoring and Control
419 [DSP0248](#), such as [DSP0257](#) and/or [DSP0267](#). Please note that firmware update only reflects the

420 conclusion of the firmware programming operation; it is device-specific whether this detection additionally
421 implies that new firmware is already active.

422 5.3.12 Accelerator clock speed sensor

423 The clock speed sensor of the Accelerator is used to read the clock speed and is represented using
424 numeric sensors. An Accelerator may have multiple clock domains, each with its own clock speed sensor

425 5.3.13 Memory temperature sensor

426 The temperature sensors on the memory modules and internal memory report the memory temperatures
427 and are represented using numeric sensors. There may be multiple memory temperature sensors
428 installed on the internal memory, on the soldered memory and on the DIMMs.

429 The memory which is soldered on the Accelerator card PCB may not have a temperature sensor on them.
430 In this case, the implementations may choose to have a temperature sensor near the soldered memory
431 chips calibrated to approximate the temperature of those memory devices.

432 5.3.14 Memory error statistics

433 The memory error statistics sensors report the memory error statistics (i.e., correctable errors and
434 uncorrectable errors) and are represented using numeric sensors. Refer to the “SensorUnits
435 enumeration” table in [DSP0248](#).

436 5.3.15 Memory composite state sensor

437 The memory composite state sensor combines sensors such as memory health state sensor, memory
438 cache state sensor, memory error state sensor, and memory redundant activity state sensor. The MC can
439 use this sensor to identify issues with the memory and to identify the specific maintenance operations that
440 it needs to perform. Refer to the “Memory-Related State Sets” table in [DSP0249](#) for all memory-related
441 sensors and their states.

442 5.4 Hierarchy description of the Accelerator card model elements

443 5.4.1 General

444 PLDM Accelerator Modeling uses a hierarchical model. Refer to section 10 PLDM associations and
445 section 11 Entity Association PDR of [DSP0248](#) to understand physical and logical associations.

446 5.4.2 Physical entities association

447 Physical association is defined in [DSP0248](#) as a method to associate components which are physically
448 connected to each other. The model uses this concept to describe the following structures:

- 449 • Content of the Accelerator card PCB
- 450 • Content of the Accelerators
- 451 • Content of the Memory Modules

452 A hierarchy entity is defined using an entity association PDR identified with a unique **ContainerID**
453 identifier parameter. The entity association PDR’s **ContainerEntityContainerID** references the PDR in
454 which the entity is contained. This entity association PDR shall also contain the contained entities defined
455 in [DSP2054](#) for the elements shown inside the purple dashed line of Figure 2.

456 Figure 3 shows an example of how an Accelerator card entity association PDR references its container
457 entity and contained entities:

458

| Accelerator card Entity Association PDR | | |
|---|----------------------------------|------------------|
| ContainerID | 100 | |
| RecordHandle | 1100 | |
| Container Entity | | |
| EntityType | 68 | Add-in card |
| EntityInstanceNumber | 1 | |
| ContainerEntityContainerID | 0 | System |
| AssociationType | | |
| | Physical to Physical containment | |
| Contained Entity - Accelerator | | |
| EntityType | 149 | Accelerator |
| EntityInstanceNumber | 1 | |
| ContainerEntityContainerID | 100 | Accelerator card |
| Contained Entity - Memory | | |
| EntityType | 66 | Memory |
| EntityInstanceNumber | 1 | |
| ContainerEntityContainerID | 100 | Accelerator card |

459 **Figure 3 – Hierarchy description using ContainerEntityContainerID referencing the Container**
 460 **Entity ContainerID**

461 **5.4.3 Logical entity association**

462 The [DSP0248](#) defines logical association as a method to associate components which collectively form a
 463 shared property yet are not physically part of the same component. This model uses logical association to
 464 describe the following structures:

465 Figure 4 shows logical association between an Accelerator and a memory module:

| Channel #1 Entity Association PDR | | |
|-----------------------------------|---------------------|---|
| ContainerID | 900 | |
| RecordHandle | 1180 | |
| Container Entity | | |
| EntityType | 79 | Processor/memory module (processor and memory together on a module) |
| EntityInstanceNumber | 1 | |
| ContainerEntityContainerID | 100 | Accelerator card |
| AssociationType | | |
| | Logical containment | |
| Contained Entity - Accelerator | | |
| EntityType | 149 | Accelerator |
| EntityInstanceNumber | 1 | |
| ContainerEntityContainerID | 100 | Accelerator card |
| Contained Entity - Memory Module | | |
| EntityType | 66 | Memory module |
| EntityInstanceNumber | 1 | |
| ContainerEntityContainerID | 100 | Accelerator card |

466 **Figure 4 – Defining a logical association**

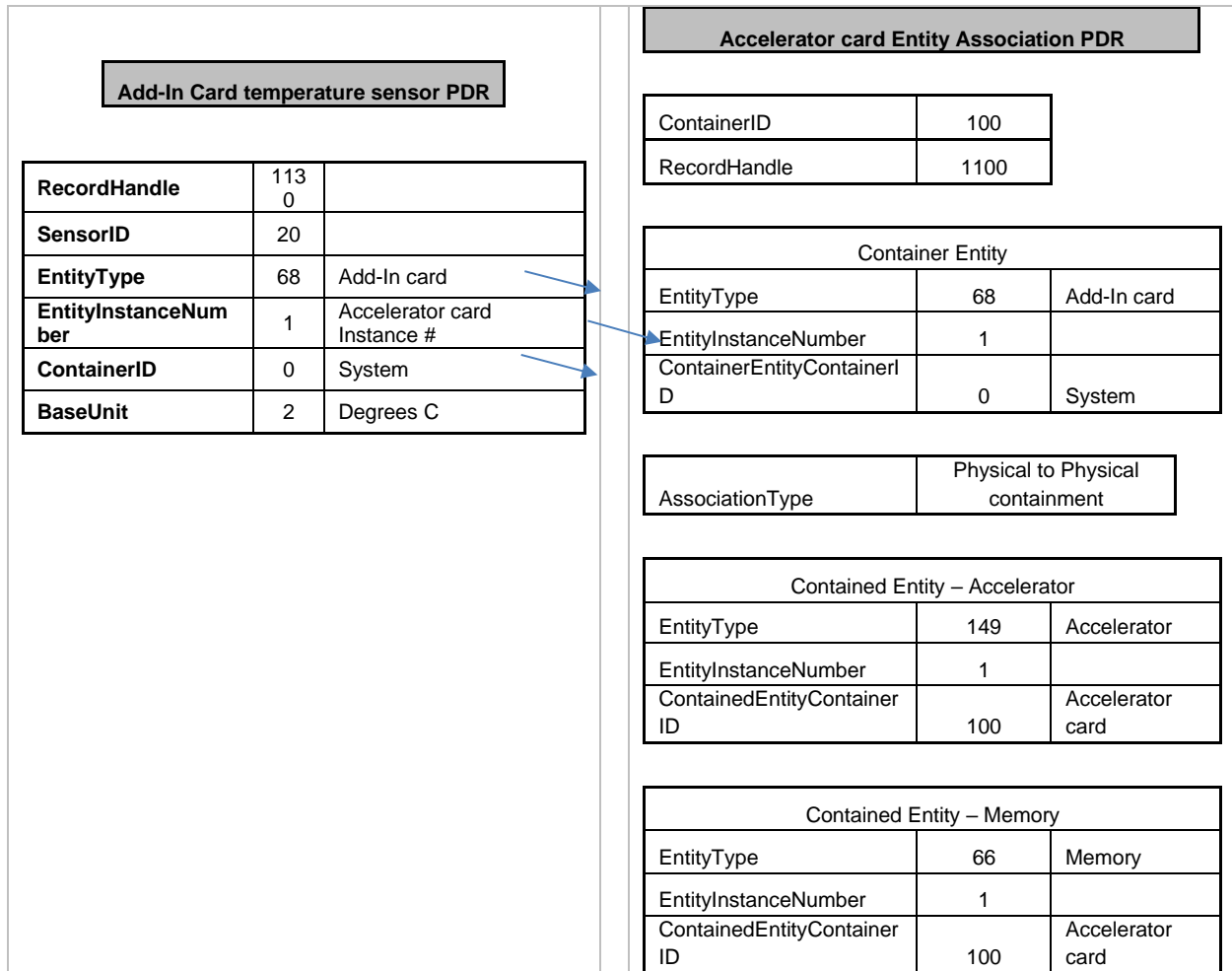
467 **5.4.4 Sensor association**

468 As per DSP0248, numeric and state sensors are not included inside entity association PDRs. They are
 469 instead associated to the measured entity by directly referencing the EntityContainerID, EntityType, and
 470 EntityInstanceNumber of the measured entity in an entity association PDR. A sensor is identified by a
 471 unique SensorID value.

472 **5.4.4.1 Associating a sensor at the top level**

473 When associating a sensor to the top-level entity which is the system the association uses the top-level
 474 ContainerEntityType containerEntityInstanceNumber and ContainerEntityContainerID parameters.

475 Figure 5 illustrates the association of a temperature sensor to the Accelerator card in the model.



476 **Figure 5 – Top-level sensor association**

477 **5.5 Element PLDM Type IDs**

478 The model uses the following Type ID for each component in the model, selected from the available types
 479 defined in [DSP0249](#). Table 1 lists the chosen Type IDs used in the model:

480 **Table 1 – Type IDs used in the Accelerator card model**

| Component | Type ID |
|------------------|---------|
| Accelerator card | 68 |
| Accelerator | 149 |
| Memory Module | 66 |

481 **5.6 Enumeration**

482 **5.6.1 General**

483 PLDM for Platform Monitoring and Control [DSP0248](#) uses enumerated IDs to define elements in the
 484 database. These IDs are labeled as:

- 485 • ContainerID – unique for each container PDR in the model database
- 486 • EntityInstanceNumber – unique for each entity type within a given hierarchy level
- 487 • RecordHandle – unique ID for each PDR in the model database
- 488 • SensorID – unique for each sensor in the model database

489 The proposed model provides an example enumeration scheme for these IDs, allowing a reasonably
 490 scalable formulation. This model is only an example and implementations should not rely on these values.

491 **5.6.2 Enumeration scheme**

492 The model assumes some maximal limits to define the enumerated values. These limits are provided as
 493 an example and can be adjusted according to the specific Accelerator card requirements.

494 The example model enumeration is designed to support an Accelerator card that does not exceed the
 495 following limits:

496 **Table 2 – Chosen enumeration limits in the model**

| Model Limit | Value |
|---|-------|
| Max Accelerators | 10 |
| Max Memory Modules | 10 |
| Max board temperature sensors | 10 |
| Max temperature sensors per Accelerator | 10 |

497 If one of the above limits is insufficient for an Accelerator card, only the enumerated values will be
 498 affected, and the model structure will not have to change.

499 Table 3 illustrates the enumeration scheme, calculated based on the above limits.

500

Table 3 – Example Enumeration Scheme with Type IDs

| Item | Max Count | Base ContainerID | Max ContainerID | Base Handle | Max Handle | Base SensorID | Max SensorID | Base Instance | Max Instance | EntityType |
|---|-----------|------------------|-----------------|-------------|------------|---------------|--------------|---------------|--------------|------------|
| Accelerator card | 1 | 100 | | 1100 | | | | 1 | 1 | 68 |
| Accelerator card Composite State Sensor | 1 | | | 1101 | 1101 | 5 | 5 | 1 | 1 | 68 |
| Accelerator card Power Sensor | 1 | | | 1102 | 1102 | 6 | 6 | 1 | 1 | 68 |
| Accelerator card Temperature sensors | 10 | | | 1130 | 1139 | 20 | 29 | 1 | 10 | 68 |
| Accelerator card fan speed sensor | 10 | | | 1150 | 1159 | 40 | 49 | 1 | 10 | 68 |
| Accelerator card Voltage sensor | 10 | | | 1170 | 1179 | 80 | 89 | 1 | 10 | 68 |
| Processor Memory Interface | 10 | 900 | 909 | 1180 | 1189 | 90 | 99 | 1 | 10 | 68 |
| Connectors | 20 | 1040 | 1059 | 1190 | 1209 | 100 | 119 | 1 | 20 | 185 |
| Memory module | 10 | 1020 | 1029 | 1210 | 1219 | | | 1 | 10 | 66 |
| Memory composite state sensor | 1 | | | 1220 | 1220 | 120 | 120 | | 1 | 66 |
| Memory temperature sensor | 20 | | | 1225 | 1244 | 125 | 144 | 1 | 20 | 66 |
| Memory module correctable Errors | 10 | | | 1255 | 1264 | 150 | 159 | | 1 | 66 |
| Memory module uncorrectable Errors | 10 | | | 1275 | 1284 | 180 | 189 | | 1 | 66 |
| Accelerators | 10 | 1000 | 1009 | 1295 | 1304 | | | 1 | 10 | 149 |
| Accelerator power sensor | 1 | | | 1310 | 1310 | 210 | 210 | | 1 | 149 |
| Accelerator State sensor | 1 | | | 1315 | 1315 | 220 | 220 | | 1 | 149 |
| Accelerator temperature sensor | 10 | | | 1325 | 1334 | 240 | 249 | 1 | 10 | 149 |
| Accelerator clock speed sensor | 10 | | | 1335 | 1344 | 260 | 269 | 1 | 10 | 149 |
| Accelerators Ports | 10 | | | 1345 | 1354 | 290 | 299 | 1 | 10 | 149 |
| Accelerators Port State | 10 | | | 1360 | 1369 | 320 | 329 | 1 | 10 | 149 |
| Accelerators Link Speed | 10 | | | 1380 | 1389 | 350 | 359 | 1 | 10 | 149 |
| Auxiliary Device Temp Sensor | 1 | | | 1395 | 1395 | 380 | 380 | | 1 | 68 |
| Auxiliary Device health sensor | 1 | | | 1400 | 1400 | 395 | 395 | | 1 | 68 |
| Plugs | 20 | 1070 | 1089 | 1410 | 1429 | 410 | 429 | 1 | 20 | 214 |
| Plug Composite Sensor | 1 | | | 1430 | 1430 | 450 | 450 | 1 | 1 | 214 |
| Plug Power Sensor | 20 | | | 1440 | 1459 | 470 | 489 | 1 | 20 | 214 |
| Plug Temp Sensor | 10 | | | 1470 | 1479 | 510 | 519 | 1 | 10 | 214 |
| Cable | 16 | | | | | | | 1 | 16 | 187 |
| Communication Channel | 100 | 800 | 899 | 1490 | 1589 | | | 1 | 100 | 79 |

501

| |
|--|
| Calculated |
| Model Constant |
| Model Sensors described in this doc |
| Common sensors for NIC and Accelerator |
| NA |

502 5.7 Model illustration

503 5.7.1 General

504 The Accelerator card PLDM model is a hierarchical model. The following subclauses describe the model
505 for each of the hierarchy levels:

506 5.7.2 Accelerator Card

507 The Accelerator card top level may contain the PCB card, Accelerators, Memory modules, one or more
508 thermal sensors, Accelerator card composite state sensor, Fan speed sensor, power sensor and voltage
509 sensors. The PCB power consumption is represented with a power sensor. The Accelerator card
510 operational state is represented by a composite state sensor. When there are multiple Accelerators on
511 the same card, Accelerator card sensors are typically only reported by the first Accelerator. The
512 Accelerator card is responsible for determining the order of accelerators in the card. Note that the top-
513 level health state sensor of the composite state sensor may reflect the card level sensors and the health
514 states of Accelerators.

515 Refer the purple dashed line in Figure 2 to the Network port link speed sensor, Network port link state
516 sensor, Pluggable module temperature sensor, pluggable module power sensor and Pluggable module
517 composite state sensor sections of [DSP2054](#) specification for networking functionality.

518 5.7.3 Accelerator

519 The Accelerator hierarchy represents the active device (or one of multiple devices) that performs the
520 Accelerator control interface. An Accelerator is represented as a collection of sensors.

521 5.7.4 Memory

522 The Memory hierarchy represents a memory device (or one of multiple devices). A Memory is
523 represented as a collection of sensors.

524 5.8 Events

525 5.8.1 General

526 This model supports using PLDM events as a method to notify the MC upon changes in the sensor
527 readings/states as described in [DSP2048](#). The following example events can be used with the model and
528 the implementation may choose to have more events.

529 5.8.2 Accelerator firmware version change

530 This event indicates to the MC that the firmware version of the Accelerator has changed. The MC may
531 use the *GetPDRRepositoryInfo* command and check if the *UpdateTime* parameter value has changed
532 since it last read the PDRs. The MC may update the whole PDR repository by re-reading all the PDRs.
533 The value used for the *UpdateTime* can be a virtual time value initialized by the Accelerator at device
534 initialization.

535 5.8.3 Health and state sensors events notifications

536 The sensors on the accelerator card may report a change in value, health, or state using a PLDM state or
537 numeric sensor event. Providing such a notification can significantly shorten the response time, compared
538 to waiting for the MC to poll the sensors, for an occurrence that requires the MC to take an action such as
539 increasing the airflow from a cooling fan.

540 **6 Model use example**

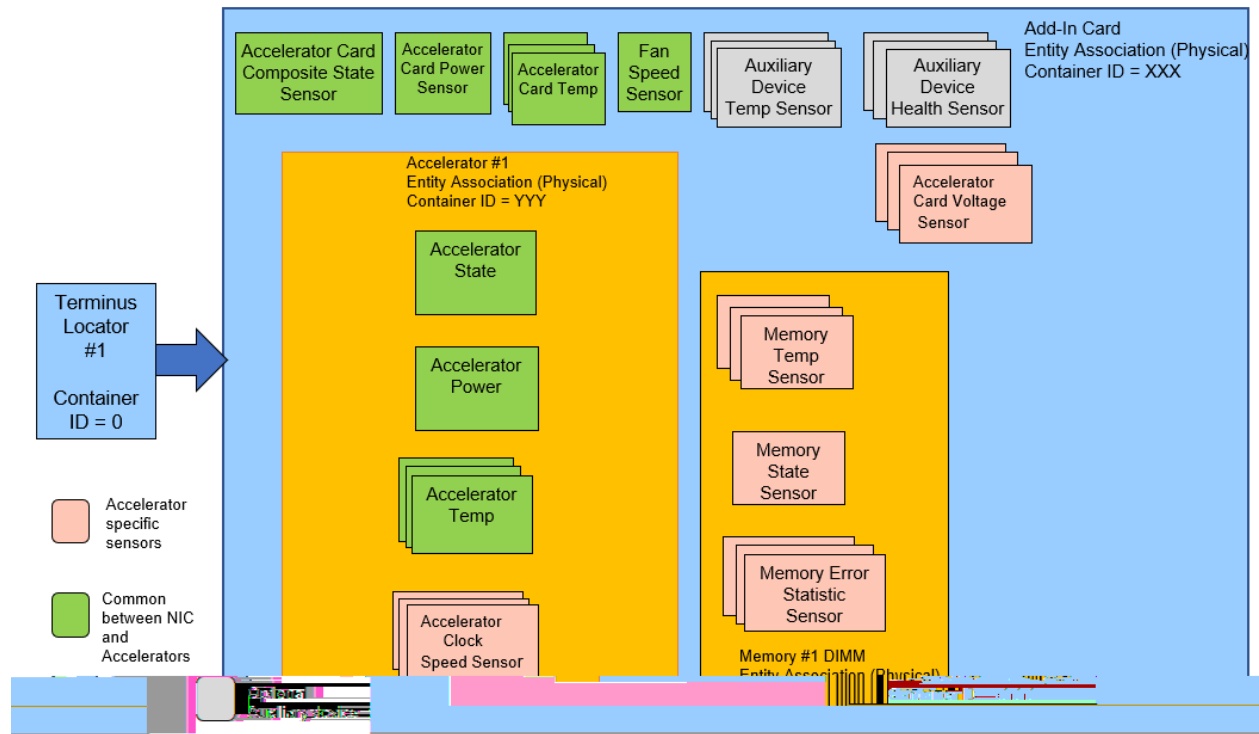
541 **6.1 General**

542 The following example for modeling an Accelerator card using PLDM for Platform Monitoring and Control
543 [DSP0248](#) describes an Accelerator card with the following attributes:

- 544 • Accelerator Card
 - 545 ○ Temperature Sensor
 - 546 ○ State Sensor
 - 547 ○ Fan speed Sensor
 - 548 ○ Voltage Sensors
 - 549 ○ Power Sensor
 - 550 ○ Auxiliary Device Temperature Sensor
 - 551 ○ Auxiliary Device Health Sensor
- 552 • Accelerator
 - 553 ○ Temperature Sensor
 - 554 ○ Power Sensor
 - 555 ○ State Sensor
 - 556 ○ Clock speed Sensor
- 557 • Memory
 - 558 ○ Temperature Sensor
 - 559 ○ Memory State Sensor
 - 560 ○ Memory Error statistics Sensor

561 Figure 6 illustrates the model which is used in the example.

562



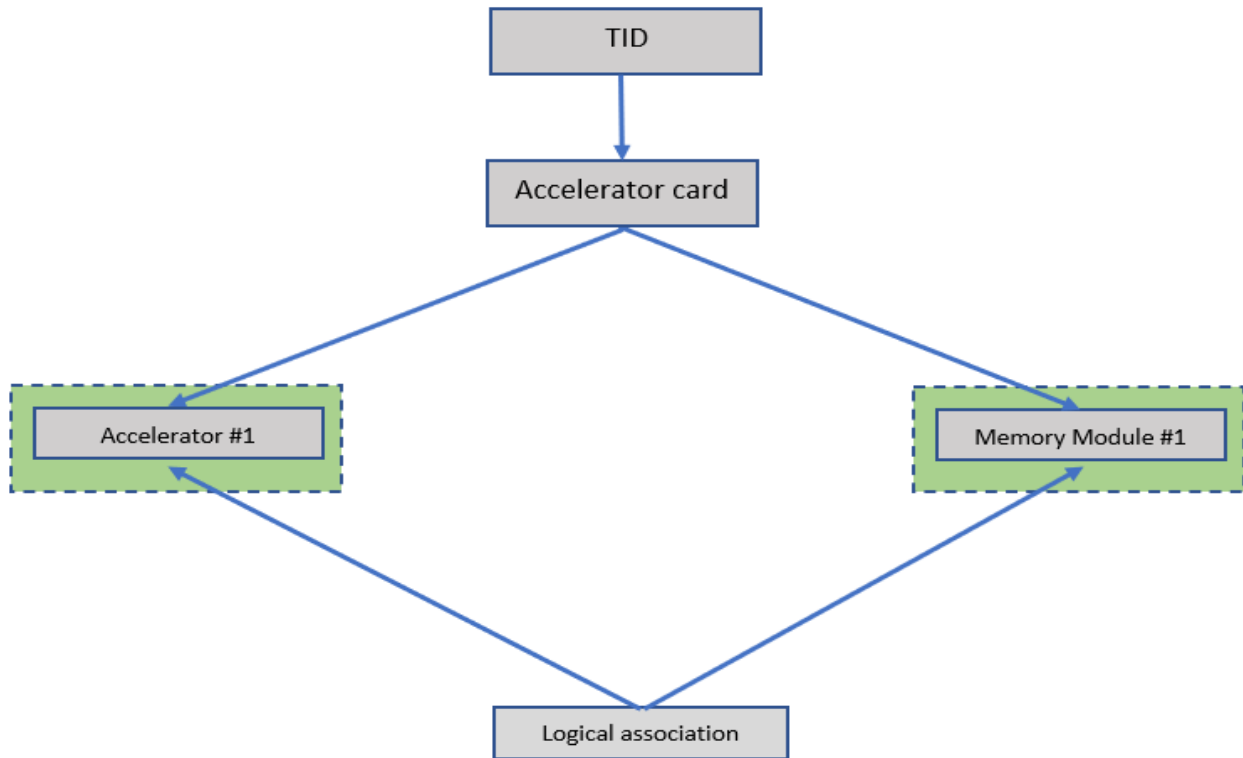
563

564

Figure 6 – Example model diagram

565 **6.2 Model hierarchy**

566 The model PDRs identify the elements depicted in Figure 6. The hierarchies are illustrated in the following
 567 diagram. For simplicity, Figure 7 shows sensors of Accelerator and Memory Module.



568

569 **Figure 7 – Accelerator card model hierarchy**

570 **6.3 Top-level TID**

571 The terminus ID is identified by the terminus locator PDR. The TID defines the top-level entry point to the
 572 PLDM model. Because there is only one Accelerator on the Accelerator card in this example, there is only
 573 one TID.

574

Table 4 – TID PDR

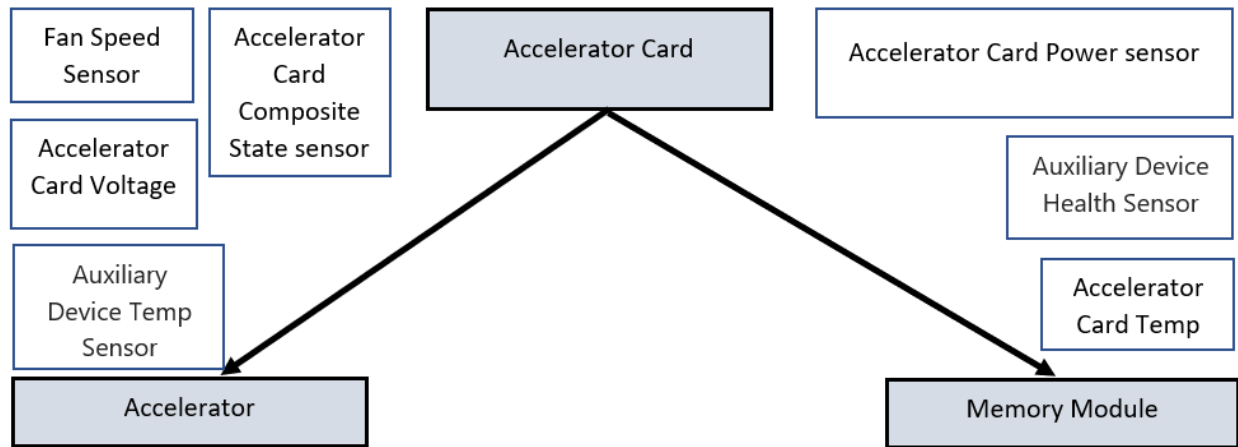
| Field name | Value | Description |
|--------------------------|-------|-----------------------------------|
| ContainerID | 0 | System |
| TID | | Assigned by MC |
| RecordHandle | 10 | Opaque number |
| TerminusLocatorValueSize | 1 | Size of (EID) or size of (UID) |
| TerminusLocatorType | 1 | MCTP EID |
| EID | EID | MCTP assigned EID Value |
| UID | UID | Vendor provided UUID format value |

575 The TID value is assigned to the terminus by the MC. When the transport layer is MCTP, the identification
 576 of the terminus is performed using the Endpoint ID (EID) value. When using PLDM
 577 over RBT, the terminus locator PDR shall use the UID (instead of EID). The UID value in the terminus
 578 locator PDR uses the device UUID value as the terminus UID. For more information regarding terminus
 579 locator PDR, see [DSP0248](#).

580 **6.4 Accelerator card**

581 **6.4.1 General**

582 The top level of the model is the Accelerator card. The Accelerator card includes the physical elements
 583 which are an Accelerator (only one Accelerator in this example) and a memory module (only one memory
 584 module in this example).



585

586 **Figure 8 – Accelerator card level elements**

587 The sensors on the Accelerator card level are described using a reference to the measured entity,
 588 independent of the container that includes all the physical elements on the Accelerator card.

| Accelerator card Entity Association PDR | | |
|---|----------------------------------|------------------|
| ContainerID | 100 | |
| RecordHandle | 1100 | |
| Container Entity | | |
| EntityType | 68 | Add-In card |
| EntityInstanceNumber | 1 | |
| ContainerEntityContainerID | 0 | System |
| AssociationType | Physical to Physical containment | |
| Contained Entity – Accelerator | | |
| EntityType | 149 | Accelerator |
| EntityInstanceNumber | 1 | |
| Contained Entity ContainerID | 100 | Accelerator card |
| Contained Entity – Memory | | |
| EntityType | 66 | Memory |
| EntityInstanceNumber | 1 | |
| Contained Entity ContainerID | 100 | Accelerator card |

589

Figure 9 – Accelerator card container PDR

590
591
592

Note that the Accelerator card ContainerID, 100, is referenced by the sensors not included in the entity association PDR. The enumeration model shown in Table 3 includes the ContainerID for every hierarchy level.

593 **6.4.2 Accelerator card power sensor**

594

| Accelerator card power sensor PDR | | |
|-----------------------------------|-------|-----------------------------|
| Field | Value | Description |
| RecordHandle | 1102 | |
| SensorID | 6 | |
| EntityType | 68 | Add-In card |
| EntityInstanceNumber | 1 | Accelerator card Instance # |
| ContainerID | 0 | System |
| BaseUnit | 7 | Watt |
| UnitModifier | -1 | 0.1 watt resolution |

595

Figure 10 – Accelerator card power sensor PDR

596 **6.4.3 Accelerator card temperature sensor**

597

| Ambient Temperature sensor PDR | | |
|--------------------------------|-------|-----------------------------|
| Field | Value | Description |
| RecordHandle | 1130 | |
| SensorID | 20 | |
| EntityType | 68 | Add-In card |
| EntityInstanceNumber | 1 | Accelerator card Instance # |
| ContainerID | 0 | System |
| BaseUnit | 2 | Degrees C |
| UnitModifier | 0 | No need for scaling |

598

Figure 11 – Ambient Temperature sensor PDR

599 **6.4.4 Accelerator card fan speed sensor**

600

| Accelerator card fan speed sensor PDR | | |
|---------------------------------------|-------|-----------------------------|
| Field | Value | Description |
| RecordHandle | 1150 | |
| SensorID | 40 | |
| EntityType | 68 | Add-In card |
| EntityInstanceNumber | 1 | Accelerator card Instance # |
| ContainerID | 0 | System |
| BaseUnit | 19 | RPM |
| UnitModifier | 0 | No need for scaling |

601

Figure 12 – Accelerator card fan speed sensor PDR

602 **6.4.5 Accelerator card voltage sensor**

603

| Voltage sensor PDR | | |
|----------------------|-------|-----------------------------|
| Field | Value | Description |
| RecordHandle | 1170 | |
| SensorID | 80 | |
| EntityType | 68 | Add-In card |
| EntityInstanceNumber | 1 | Accelerator card Instance # |
| ContainerID | 0 | System |
| BaseUnit | 5 | Volts |
| UnitModifier | -1 | 0.1 volt resolution |

604

Figure 13 – Accelerator card voltage sensor PDR

605 **6.4.6 Accelerator card auxiliary device temperature sensor**

606

| Auxiliary device temperature sensor PDR | | |
|---|-------|-----------------------------|
| Field | Value | Description |
| RecordHandle | 1395 | |
| SensorID | 380 | |
| EntityType | 68 | Add-In card |
| EntityInstanceNumber | 1 | Accelerator card Instance # |
| ContainerID | 0 | System |
| BaseUnit | 2 | Degrees C |
| UnitModifier | 0 | No need for scaling |

607

Figure 14 – Auxiliary device temperature sensor PDR

608 **6.4.7 Accelerator card auxiliary device health sensor**

609

| Auxiliary device health sensor PDR | | |
|------------------------------------|---|-----------------------------|
| Field | Value | Description |
| RecordHandle | 1400 | |
| SensorID | 395 | |
| EntityType | 68 | Add-In card |
| EntityInstanceNumber | 1 | Accelerator card Instance # |
| ContainerID | 0 | System |
| StateSetID | 1 | Health state |
| PossibleStates | Refer to the “General state sets” table in DSP0249. | |

610

Figure 15 – Auxiliary device health sensor PDR

611 **6.4.8 Accelerator card composite state sensor**

612

| Accelerator card composite state sensor PDR | | |
|---|------|-------------|
| RecordHandle | 1101 | |
| EntityType | 68 | Add-In card |
| EntityInstanceNumber | 1 | |
| ContainerEntityContainerID | 0 | System |

| | |
|----------------------|---|
| PLDMTerminusHandle | 0 |
| SensorID | 5 |
| CompositeSensorCount | 3 |

| | | |
|----------------|---|--------------|
| StateSetID | 1 | Health state |
| PossibleStates | Refer to the "General state sets" table in DSP0249. | |

| | | |
|----------------|---|--------------|
| StateSetID | 21 | Thermal Trip |
| PossibleStates | Refer to the "General state sets" table in DSP0249. | |

| | | |
|----------------|---|---------------------------------|
| StateSetID | 10 | Memory Operational Fault status |
| PossibleStates | Refer to the "General state sets" table in DSP0249. | |

613

Figure 16 – Accelerator card composite state sensor PDR

614 **6.5 Accelerator**

615 **6.5.1 General**

616 The Accelerator is an active device and being a physical entity that doesn't include other entities, the
 617 Accelerator is not declared in its own PDR. It is instead declared in the Accelerator card container PDR.
 618 The Accelerator includes a set of device-level sensors. The following diagram illustrates the model
 619 sensors in the Accelerator:



620

621 **Figure 17 – Example model Accelerator**

622 The Accelerator content is declared using an entity-association PDR that includes the hierarchical
 623 description of the Accelerator. The device-level sensors are declared with separate PDRs using direct
 624 references to the measured entities.

625

| Accelerator Entity Association PDR | | |
|------------------------------------|----------------------------------|------------------|
| ContainerID | 1000 | |
| RecordHandle | 1295 | |
| Container Entity | | |
| EntityType | 149 | Accelerator |
| EntityInstanceNumber | 1 | |
| ContainerEntityContainerID | 100 | Accelerator card |
| AssociationType | Physical to Physical containment | |

626

Figure 18 – Accelerator entity association PDR

627 **6.5.2 Accelerator temperature sensor**

628

| Accelerator temperature sensor PDR | | |
|------------------------------------|-------|------------------------|
| Field | Value | Description |
| RecordHandle | 1325 | |
| SensorID | 240 | |
| EntityType | 149 | Accelerator |
| EntityInstanceNumber | 1 | Accelerator Instance # |
| ContainerID | 100 | Accelerator card |
| BaseUnit | 2 | Degrees C |

629

Figure 19 – Accelerator temperature sensor PDR

630 **6.5.3 Accelerator power sensor**

631

| Accelerator power sensor PDR | | |
|------------------------------|-------|------------------------|
| Field | Value | Description |
| RecordHandle | 1310 | |
| SensorID | 210 | |
| EntityType | 149 | Accelerator |
| EntityInstanceNumber | 1 | Accelerator Instance # |
| ContainerID | 100 | Accelerator card |
| BaseUnit | 7 | Watts |
| UnitModifier | -1 | 0.1 watt resolution |

632

Figure 20 – Accelerator power sensor PDR

633 **6.5.4 Accelerator composite state sensor**

634

| Accelerator composite state sensor PDR | | |
|--|------|------------------|
| RecordHandle | 1315 | |
| EntityType | 149 | Accelerator |
| EntityInstanceNumber | 1 | |
| ContainerEntityContainerID | 100 | Accelerator card |

| | |
|----------------------|-----|
| PLDMTerminusHandle | 0 |
| SensorID | 220 |
| CompositeSensorCount | 5 |

| | | |
|----------------|---|--------------|
| StateSetID | 1 | Health state |
| PossibleStates | Refer to the "General state sets" table in DSP0249. | |

| | | |
|----------------|---|--------------|
| StateSetID | 21 | Thermal Trip |
| PossibleStates | Refer to the "General state sets" table in DSP0249. | |

| | | |
|----------------|---|------------------|
| StateSetID | 18 | Firmware Version |
| PossibleStates | Refer to the "General state sets" table in DSP0249. | |

| | | |
|----------------|---|---------------|
| StateSetID | 15 | Configuration |
| PossibleStates | Refer to the "General state sets" table in DSP0249. | |

| | | |
|----------------|---|----------------------|
| StateSetID | 16 | Configuration Change |
| PossibleStates | Refer to the "General state sets" table in DSP0249. | |

635

Figure 21 – Accelerator composite state sensor PDR

636 **6.5.5 Accelerator clock speed sensor**

637

| Accelerator clock speed sensor PDR | | |
|------------------------------------|-------|------------------------|
| Field | Value | Description |
| RecordHandle | 1335 | |
| SensorID | 260 | |
| EntityType | 149 | Accelerator |
| EntityInstanceNumber | 1 | Accelerator Instance # |
| ContainerID | 100 | Accelerator Card |
| BaseUnit | 20 | Hertz |
| UnitModifier | 6 | 1 MHz resolution |

638

Figure 22 – Accelerator card clock speed sensor PDR

639 **6.6 Memory**

640 **6.6.1 General**

641 The Memory is a physical entity in the model. The Memory is already declared within the Accelerator card
 642 container PDR. The Memory includes a set of device-level sensors. The Memory sensors cover all three
 643 types of memory i.e., DIMM, Internal memory, and soldered memory chips. The following diagram
 644 illustrates the model sensors in the Memory:

645

646

647

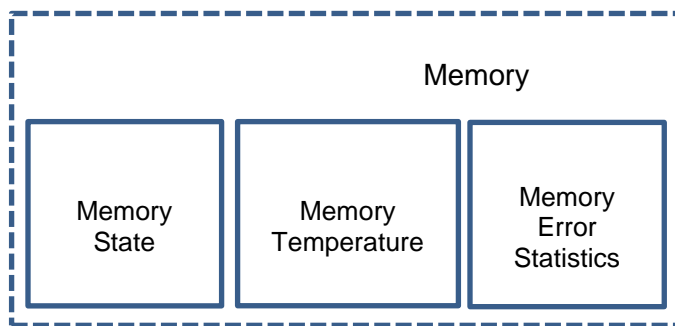
648

649

650

651

652



653

Figure 23 – Example Memory model

654 The Memory content is declared using an entity-association PDR that includes the hierarchical
 655 description of the Memory. The device-level sensors are declared with separate PDRs using direct
 656 references to the measured entities.

| Memory Association PDR | | |
|----------------------------|----------------------------------|------------------|
| ContainerID | 1020 | |
| RecordHandle | 1210 | |
| Container Entity | | |
| EntityType | 66 | Memory |
| EntityInstanceNumber | 1 | |
| ContainerEntityContainerID | 100 | Accelerator card |
| AssociationType | Physical to Physical containment | |

657 **Figure 24 – Memory association PDR**

658 **6.6.2 Memory temperature sensor**

659

| Memory temperature sensor PDR | | |
|-------------------------------|-------|-------------------|
| Field | Value | Description |
| RecordHandle | 1225 | |
| SensorID | 125 | |
| EntityType | 66 | Memory |
| EntityInstanceNumber | 1 | Memory Instance # |
| ContainerID | 100 | Accelerator card |
| BaseUnit | 2 | Degrees C |

660 **Figure 25 – Memory temperature sensor PDR**

661 **6.6.3 Memory error statistics sensors**

662

| Memory correctable errors PDR | | |
|-------------------------------|-------|--------------------|
| Field | Value | Description |
| RecordHandle | 1255 | |
| SensorID | 150 | |
| EntityType | 66 | Memory |
| EntityInstanceNumber | 1 | Memory instance # |
| ContainerID | 100 | Accelerator card |
| BaseUnit | 80 | Correctable Errors |

663 **Figure 26 – Memory correctable errors PDR**

664

| Memory uncorrectable errors PDR | | |
|---------------------------------|-------|----------------------|
| Field | Value | Description |
| RecordHandle | 1275 | |
| SensorID | 180 | |
| EntityType | 66 | Memory |
| EntityInstanceNumber | 1 | Memory Instance # |
| ContainerID | 100 | Accelerator card |
| BaseUnit | 81 | Uncorrectable Errors |

665

Figure 27 – Memory uncorrectable errors PDR

666 **6.6.4 Memory composite state sensor**

667

| Memory composite state sensor PDR | | |
|-----------------------------------|------|------------------|
| RecordHandle | 1220 | |
| EntityType | 66 | Memory |
| EntityInstanceNumber | 1 | |
| ContainerEntityContainerID | 100 | Accelerator card |

| | |
|----------------------|-----|
| PLDMTerminusHandle | 0 |
| SensorID | 120 |
| CompositeSensorCount | 4 |

| | | |
|----------------|--|--------------|
| StateSetID | 1 | Health state |
| PossibleStates | Refer to the “General state sets” table in DSP0249 . | |

| | | |
|----------------|---|---------------------|
| StateSetID | 320 | Memory cache status |
| PossibleStates | Refer to the “Memory-Related State Sets” table in DSP0249 . | |

| | | |
|----------------|---|---------------------|
| StateSetID | 321 | Memory error status |
| PossibleStates | Refer to the “Memory-Related State Sets” table in DSP0249 . | |

| | | |
|----------------|---|----------------------------------|
| StateSetID | 322 | Redundant Memory activity status |
| PossibleStates | Refer to the “Memory-Related State Sets” table in DSP0249 . | |

668

Figure 28 – Memory composite state sensor PDR

ANNEX A (informative)

669
670
671
672
673

Notation and conventions

674 A.1 Notations

675 Examples of notations used in this document are as follows:

- 676 • 2:N In field descriptions, this will typically be used to represent a range of byte offsets
677 starting from byte two and continuing to and including byte N. The lowest offset is on
678 the left; the highest is on the right.
- 679 • (6) Parentheses around a single number can be used in message field descriptions to
680 indicate a byte field that may be present or absent.
- 681 • (3:6) Parentheses around a field consisting of a range of bytes indicates the entire range
682 may be present or absent. The lowest offset is on the left; the highest is on the right.
- 683 • PCIe Underlined, blue text is typically used to indicate a reference to a document or
684 specification called out in "Normative references" clause or to items hyperlinked within
685 the document.
- 686 • rsvd This case-insensitive abbreviation is for "reserved."
- 687 • [4] Square brackets around a number are typically used to indicate a bit offset. Bit offsets
688 are given as zero-based values (that is, the least significant bit [LSb] offset = 0).
- 689 • [7:5] This notation indicates a range of bit offsets. The most significant bit is on the left; the
690 least significant bit is on the right.
- 691 • 1b The lowercase "b" following a number consisting of 0s and 1s is used to indicate the
692 number is being given in binary format.
- 693 • 0x12A A leading "0x" is used to indicate a number given in hexadecimal format.

694

695
696
697
698
699

ANNEX B (informative)

Change log

| Version | Date | Description |
|---------|------------|--|
| 1.0.0 | 2022-05-25 | Initial draft |
| 1.0.1 | 2024-09-19 | Fixed case to CamelCase Fixed some names different than DSP0248: <ul style="list-style-type: none"> • entityInstance → EntityInstanceNumber • SensorType → StateSetID • TerminusHandle → PLDMTerminusHandle • TerminusLocatorSize → TerminusLocatorValueSize • InstanceID → EntityInstanceNumber • HandleID → RecordHandle Fixed the TID PDR RecordHandle to 10 |

700